Dictionnaire de Bitcoin

Dictionnaire de Bitcoin

TOUT LE VOCABULAIRE TECHNIQUE DE BITCOIN

Loïc Morel

© 2026 Loïc Morel

Dictionnaire de Bitcoin : Tout le vocabulaire technique de Bitcoin

Version du 23 mars 2026
https://github.com/LoicPandul/Dictionnaire-de-Bitcoin/

Cet ouvrage est sous licence CC BY-NC-SA 4.0
https://creativecommons.org/licenses/by-nc-sa/4.0/

Lightning : sats@pandul.fr
Email : loic@pandul.fr
Site web : https://pandul.fr/
GitHub : https://github.com/LoicPandul/

ISBN : 9798250570664

CONTRIBUTEURS ET REMERCIEMENTS

Merci à tous les contributeurs pour leur aide, leurs conseils, la rédaction de nouvelles définitions, la correction des entrées existantes et la relecture de l'ouvrage :

- Psyduck07 (https://github.com/Psyduck07) ;
- Adrien Lacombe (https://github.com/adrienlacombe) ;
- Ludovic Lars (https://github.com/lugaxker) ;
- Beemo (https://github.com/nflatrea) ;
- quinoah (https://github.com/quinoah).

Dictionnaire de Bitcoin est un projet participatif ouvert à tous : vous pouvez proposer des modifications, des corrections ou de nouvelles définitions via le dépôt GitHub : https://github.com/LoicPandul/Dictionnaire-de-Bitcoin/.

Merci également à toutes les personnes qui m'ont aidé dans la rédaction de mes autres contenus, qui ont servi de fondement au *Dictionnaire de Bitcoin* :

- Grittoshi (https://x.com/Grittoshi) ;
- 200KEKS (https://x.com/200KEKS) ;
- Fanis Michalakis (https://x.com/FanisMichalakis) ;
- AcidBunny (https://x.com/acidbunny21) ;
- Sosthène (https://x.com/Sosthene___) ;
- Théo Pantamis (https://x.com/TheoPantamis) ;
- Louferlou (https://x.com/Louferlou) ;
- JohnOnChain (https://x.com/JohnOnChain) ;
- Pythcoiner (https://x.com/pythcoiner) ;
- Rogzy (https://x.com/DecouvreBitcoin) ;
- Théo Mogenet (https://x.com/theomogenet) ;
- Gilles Cadignan (https://x.com/gillesCadignan) ;
- Ludovic Lars (https://x.com/lugaxker) ;
- Lounès Ksouri (https://x.com/louneskmt) ;
- WillKEKS (https://x.com/Satsmokers) ;
- Paul ADW (https://x.com/PaulADW) ;
- TDevD (https://x.com/SamouraiDev) ;
- Trigger (https://x.com/Trigger_jw) ;
- GégéLSMR (https://x.com/gegelsmr4) ;
- Bitcoiner Nomad (https://x.com/BitcoinerNomad) ;
- LaurentMT (https://x.com/LaurentMT) ;
- Guillaume Goualard (https://x.com/_INITIO_) ;
- Science (https://x.com/science_genial) ;
- Marc VALLEE (https://x.com/marcvallee13) ;
- Meyga Vox (https://x.com/meygavox) ;
- Mill3sim3 (https://x.com/Mill3sim3) ;
- Jean-Luc et Marco de Bitcoin.fr (https://bitcoin.fr/) ;
- L'équipe de Découvre Bitcoin (https://decouvrebitcoin.fr/) ;
- L'équipe de Plan ₿ Academy (https://planb.academy/) ;

- L'équipe de Bitstack (https://www.bitstack-app.com/) ;
- L'équipe de Parlons Bitcoin (https://x.com/parlonsbitcoin).

Un grand merci aussi à tous ceux qui ont contribué à mes différents projets et qui ont choisi de rester anonymes.

Merci également à toutes les personnes qui créent des contenus éducatifs sur Bitcoin et son écosystème. Je n'ai pas cité de sources directement, mais les nombreux contenus que j'ai consultés au fil des années ont enrichi mes connaissances et ont forcément inspiré la rédaction de ce dictionnaire. Un remerciement spécial à ceux que je lis ou écoute régulièrement :

- Andreas Antonopoulos (https://x.com/aantonop) ;
- Sjors Provoost (https://x.com/provoost) ;
- Eric Voskuil (https://x.com/evoskuil) ;
- Fanis Michalakis (https://x.com/FanisMichalakis) ;
- Ludovic Lars (https://x.com/lugaxker) ;
- JohnOnChain (https://x.com/JohnOnChain) ;
- Théo Pantamis (https://x.com/TheoPantamis) ;
- LaurentMT (https://x.com/LaurentMT) ;
- Théo Mogenet (https://x.com/theomogenet) ;
- Sosthène (https://x.com/Sosthene___) ;
- Lounès Ksouri (https://x.com/louneskmt) ;
- Satoshi Nakamoto ;
- Et plein d'autres que j'oublie sûrement.

Je suis également profondément reconnaissant envers tous ceux qui suivent mes travaux, les lisent et les partagent, sur les réseaux sociaux comme ailleurs. Vos partages spontanés sont pour moi une source de motivation.

Et évidemment, un immense merci à ma femme, Chloé, qui est à l'origine de l'idée de ce dictionnaire et qui m'a conseillé tout au long de sa rédaction.

Si vous souhaitez soutenir le projet, vous pouvez acheter la version physique sur amazon.fr et bitcoinbazar.fr, ou bien me faire un don sur mon adresse Lightning : sats@pandul.fr. Merci !

Le 24 avril 2024, les deux cofondateurs de Samourai Wallet ont été injustement arrêtés pour avoir simplement écrit du code. Je tiens à leur exprimer mon soutien indéfectible. Leur engagement pour la protection de la vie privée et la liberté incarne les valeurs fondamentales de Bitcoin. Ces développeurs n'ont commis aucun crime ; ils ont seulement œuvré pour offrir des outils permettant à chacun de faire valoir ses droits naturels. Afin de les soutenir dans cette épreuve, j'invite chacun à signer la pétition en ligne : https://billandkeonne.org/.

NOTE DE L'AUTEUR

Le *Dictionnaire de Bitcoin* recense et définit les termes techniques liés à Bitcoin et son écosystème.

GUIDE DE LECTURE

Chaque définition suit le format suivant :

- Titre : Le terme défini ;
- Catégorie : Le domaine concerné (Cryptographie, Réseau, Portefeuille, etc.) ;
- Traduction : L'équivalent anglais ou français du terme ;
- Définition : L'explication détaillée du concept ;
- Renvois : Les termes associés à cette définition.

Pour une lecture efficace, n'hésitez pas à suivre les termes associés : ils vous permettront de relier les notions entre elles et de progresser de définition en définition.

CONTRIBUTION

Dictionnaire de Bitcoin est un projet ouvert. Vous êtes libre de proposer tout type de modification ou de correction, ce qui inclut notamment :

- La suggestion de nouveaux termes à ajouter (avec ou sans définition) ;
- La correction ou l'amélioration d'une définition qui vous semble erronée ou imprécise ;
- La correction d'une faute d'orthographe ou d'une coquille ;
- Toute autre suggestion de modification.

Pour contribuer, vous pouvez proposer une pull request ou bien ouvrir une issue sur le dépôt GitHub du projet (https://github.com/LoicPandul/Dictionnaire-de-Bitcoin/). Pour plus de détails sur les différentes manières de participer, référez-vous au fichier README de ce dépôt.

PRINCIPES DE RÉDACTION

Publier un dictionnaire implique de rédiger des définitions, et donc d'opérer des choix quant aux termes à inclure. Le parti pris que j'ai adopté consiste à intégrer tout terme dès lors qu'il entretient un lien avec Bitcoin ou son écosystème. L'ambition de ce dictionnaire est de couvrir le champ lexical de Bitcoin avec le plus haut degré d'exhaustivité possible, dans l'esprit des dictionnaires professionnels propres à d'autres disciplines.

Pour simplifier, un terme est ajouté si la réponse à la question suivante est affirmative : « *Un débutant qui étudie Bitcoin pourrait-il croiser ce terme au cours de ses recherches et souhaiter en comprendre la signification ?* »

Cela englobe l'ensemble des termes techniques, des logiciels, des algorithmes et des protocoles liés, directement ou indirectement, à Bitcoin ou à des protocoles qui s'appuient sur Bitcoin (comme Lightning, RGB, Liquid, Ark, etc.).

Les termes plus généraux relevant de l'informatique ou de la cryptographie ne sont inclus que s'ils sont susceptibles d'être rencontrés et s'ils sont nécessaires à la compréhension de Bitcoin.

Enfin, vous remarquerez que je ne me suis imposé aucune contrainte de longueur pour les définitions. Je préfère avoir une définition exhaustive, même si elle est longue, à une définition courte mais imprécise.

Concernant les noms propres, j'ai fait les choix suivants :

- Pour les personnes, je les ai intégrées comme entrées indépendantes uniquement lorsqu'elles ont joué un rôle historique et majeur dans le développement de Bitcoin (Satoshi Nakamoto ou Hal Finney, par exemple). Pour les autres personnalités importantes de l'écosystème, je préfère les mentionner au sein de la définition de leur contribution (Wei Dai dans la définition de b-money, par exemple) ;
- Pour les organisations (entreprises, marques, projets, associations…), j'ai choisi de les inclure comme définitions indépendantes uniquement lorsqu'il s'agit d'entités que je considère comme majeures. La même logique s'applique aux logiciels et aux matériels cités. L'écosystème Bitcoin est particulièrement complexe et difficile à compartimenter : les concepts sont liés à des logiciels, eux-mêmes souvent associés à des entités, et la connaissance de ces entités peut être utile pour une compréhension globale de Bitcoin. Il me semble donc cohérent d'adopter cette approche.

Je reste bien entendu ouvert aux suggestions, qu'il s'agisse d'ajouter, de modifier ou de supprimer des définitions : n'hésitez pas à m'en faire part sur GitHub via une issue ou une pull request.

AVERTISSEMENT

Ce dictionnaire a une vocation purement informative et pédagogique. Les informations qu'il contient ne constituent en aucun cas des conseils financiers, juridiques ou d'investissement. Je décline toute responsabilité quant à l'utilisation qui pourrait être faite de ces informations.

Bitcoin est un domaine en constante évolution. Malgré le soin apporté à la rédaction de cet ouvrage, certaines informations peuvent rapidement devenir obsolètes. N'hésitez pas à consulter la version en ligne la plus récente et à vérifier les informations à partir de sources officielles et de la documentation technique.

TABLE DES MATIÈRES

A

ACINQ 2
ACTIF AU PORTEUR 2
ADAPTOR SIGNATURE 2
ADDR 4
ADDR.DAT 4
ADDRESS SPOOFING 4
ADDRMAN 5
ADDRV2 5
ADRESSE DE RÉCEPTION 5
ADRESSE STATIQUE 7
AEZEED 7
AGORISME 8
AIR COOLING 8
AIR-GAPPED 8
AJUSTEMENT DE LA DIFFICULTE 9
ALBY 9
ALBY HUB 10
ALERT SYSTEM 10
ALGORITHME 11
ALTCOIN 11
ALUVM 11
AMBOSS 11
AML - ANTI-MONEY LAUNDERING 11
ANALYSE DE CHAINE 12
ANCESTOR MINING 13
ANCHOR 13
ANCHOR OUTPUTS 13
ANCHORS.DAT 14
ANCRAGE BILATÉRAL 14
ANNEX 14
ANONSETS - ANONYMITY SETS 15
ANTI-EXFIL SIGNING 16
ANTPOOL 16
AOPP 17
APERTURE 17
API 18
APO - ANYPREVOUT 18
AQUA 18
ARBITRAGE 18
ARBRE DE MERKLE 19
ARK 20
ASCII 20
ASHIGARU 21
ASIC 21
ASIC RESISTANCE 21
ASICBOOST 22
ASMAP 22
ASP - ARK SERVICE PROVIDER 23
ASSET ID 23
ASSET MERGE 24
ASSET PROOF 24
ASSET SPLIT 24
ASSIGNMENT 24
ASSUME UTXO 25
ASSUME VALID 25
ASYNC PAYMENTS 26
ATH - ALL-TIME HIGH 26
ATLC 26
ATOMIC MULTI-PATH PAYMENTS 27
ATOMIC SWAP 27
ATTAQUE DE FINNEY 28
ATTAQUE DES 51% 28
AUTOLOOP 29
AVERAGE ROUND DURATION 29

B

B-MONEY 32
BANLIST.DAT 32
BANLIST.JSON 32
BARE-MULTISIG 32
BASE - ARITHMÉTIQUE 33
BASE FEE 33
BASE58CHECK 33
BATCHED SPENDING 34
BBQR 35
BCH - BITCOIN CASH 35
BCH CODE 35
BCOIN 36
BDK - BITCOIN DEV KIT 36
BEAR MARKET 36
BECH32 ET BECH32M 37
BEHIND-THE-METER 37
BERKELEYDB 38
BETTERHASH 38
BFT - BYZANTINE FAULT TOLERANCE 38
BGP HIJACKING 39
BIG BLOCKERS 39
BIG-ENDIAN 39
BINAIRE 40
BINOHASH 40
BIP 40
BIP-0001 42

BIP-0002 43
BIP-0003 43
BIP-0008 44
BIP-0009 44
BIP-0010 45
BIP-0011 45
BIP-0012 46
BIP-0013 46
BIP-0014 46
BIP-0015 47
BIP-0016 47
BIP-0017 47
BIP-0018 48
BIP-0019 48
BIP-0020 48
BIP-0021 48
BIP-0022 49
BIP-0023 49
BIP-0030 49
BIP-0031 50
BIP-0032 50
BIP-0033 50
BIP-0034 51
BIP-0035 51
BIP-0036 51
BIP-0037 52
BIP-0038 52
BIP-0039 53
BIP-0042 53
BIP-0043 54
BIP-0044 54
BIP-0045 55
BIP-0046 56
BIP-0047 56
BIP-0048 57
BIP-0049 57
BIP-0050 57
BIP-0052 57
BIP-0053 58
BIP-0054 58
BIP-0060 59
BIP-0061 59
BIP-0062 59
BIP-0064 60
BIP-0065 60
BIP-0066 60
BIP-0067 60
BIP-0068 61
BIP-0069 61
BIP-0070 61
BIP-0071 62
BIP-0072 62
BIP-0073 62
BIP-0074 63
BIP-0075 63
BIP-0077 63
BIP-0078 64
BIP-0079 64
BIP-0080 64
BIP-0081 65
BIP-0083 65
BIP-0084 65
BIP-0085 65
BIP-0086 66
BIP-0087 66
BIP-0088 66
BIP-0089 66
BIP-0090 67
BIP-0091 67
BIP-0093 68
BIP-0094 68
BIP-0098 68
BIP-0099 68
BIP-0100 69
BIP-0101 69
BIP-0102 69
BIP-0103 69
BIP-0104 70
BIP-0105 70
BIP-0106 70
BIP-0107 70
BIP-0109 71
BIP-0110 71
BIP-0111 71
BIP-0112 72
BIP-0113 72
BIP-0114 72
BIP-0115 73
BIP-0116 73
BIP-0117 73
BIP-0118 74
BIP-0119 74
BIP-0120 74
BIP-0121 75
BIP-0122 75
BIP-0123 75
BIP-0124 76
BIP-0125 76
BIP-0126 76
BIP-0127 76
BIP-0129 77
BIP-0130 77
BIP-0131 77
BIP-0132 77
BIP-0133 78
BIP-0134 78
BIP-0135 78
BIP-0136 78
BIP-0137 79
BIP-0140 79
BIP-0141 79
BIP-0142 79

BIP-0143 80
BIP-0144 80
BIP-0145 80
BIP-0146 80
BIP-0147 81
BIP-0148 81
BIP-0149 82
BIP-0150 82
BIP-0151 83
BIP-0152 83
BIP-0154 83
BIP-0155 84
BIP-0156 84
BIP-0157 85
BIP-0158 85
BIP-0159 86
BIP-0171 86
BIP-0172 86
BIP-0173 86
BIP-0174 87
BIP-0175 87
BIP-0176 87
BIP-0177 88
BIP-0178 88
BIP-0179 88
BIP-0180 88
BIP-0197 89
BIP-0199 89
BIP-0300 89
BIP-0301 90
BIP-0310 90
BIP-0320 90
BIP-0321 91
BIP-0322 91
BIP-0324 91
BIP-0325 92
BIP-0326 92
BIP-0327 93
BIP-0328 93
BIP-0329 93
BIP-0330 94
BIP-0331 94
BIP-0337 94
BIP-0338 94
BIP-0339 95
BIP-0340 95
BIP-0341 95
BIP-0342 96
BIP-0343 96
BIP-0345 96
BIP-0346 97
BIP-0347 97
BIP-0348 97
BIP-0349 98
BIP-0350 98
BIP-0351 98
BIP-0352 99
BIP-0353 99
BIP-0360 100
BIP-0370 100
BIP-0371 100
BIP-0372 100
BIP-0373 101
BIP-0374 101
BIP-0375 101
BIP-0379 101
BIP-0380 102
BIP-0381 102
BIP-0382 102
BIP-0383 102
BIP-0384 103
BIP-0385 103
BIP-0386 103
BIP-0387 103
BIP-0388 104
BIP-0389 104
BIP-0390 104
BIP-0431 105
BIP-0433 105
BIP-0434 105
BIP-0443 106
BIRTH DATE 106
BISQ 106
BIT 107
BIT GOLD 107
BIT - UNITÉ 107
BITAXE 108
BITBANANA 108
BITBOX 108
BITCOIN ATM 108
BITCOIN - B MAJUSCULE 109
BITCOIN - B MINUSCULE 109
BITCOIN BEACH 109
BITCOIN CLASSIC 110
BITCOIN-CLI 110
BITCOIN.CONF 110
BITCOIN CORE 111
BITCOIN CORE GUI-QML 111
BITCOIN-DEV 112
BITCOIN FOG 112
BITCOIN INQUISITION 112
BITCOIN JESUS 113
BITCOIN KEEPER 113
BITCOIN KNOTS 113
BITCOIN MAGAZINE 113
BITCOIN OPTECH 114
BITCOIN QT 114
BITCOIN TREASURY COMPANY 114
BITCOIN UNLIMITED 115
BITCOIN XT 115
BITCOIND 115
BITCOIND.PID 116

BITCOINJ ... 116
BITCOINTALK ... 116
BITKEY ... 116
BITKIT ... 117
BITMAIN ... 117
BITREFILL ... 117
BITVM ... 118
BLIND ORACLE ... 118
BLINDED PATHS ... 119
BLINK ... 119
BLIP ... 119
BLIP-0001 ... 120
BLIP-0002 ... 120
BLIP-0003 ... 120
BLIP-0004 ... 121
BLIP-0010 ... 121
BLIP-0011 ... 121
BLIP-0017 ... 122
BLIP-0025 ... 122
BLIP-0032 ... 122
BLIP-0039 ... 123
BLIP-0050 ... 123
BLIP-0051 ... 123
BLIP-0052 ... 124
BLIP-0055 ... 124
BLIXT WALLET ... 124
BLK*.DAT ... 124
BLKINDEX.DAT ... 125
BLKTREE/ ... 125
BLOC ... 125
BLOCK-RELAY-ONLY ... 126
BLOCK TEMPLATE ... 126
BLOCKCHAIN ... 126
BLOCKS INDEX ... 127
BLOCKS/REV*.DAT ... 127
BLOCKSIGNERS ... 127
BLOCKSIZE WAR ... 127
BLOCKSTREAM ... 128
BLOOM FILTER ... 129
BLUE WALLET ... 129
BOLT ... 129
BOLT-00 ... 130
BOLT-01 ... 130
BOLT-02 ... 130
BOLT-03 ... 130
BOLT-04 ... 131
BOLT-05 ... 131
BOLT-07 ... 131
BOLT-08 ... 132
BOLT-09 ... 132
BOLT-10 ... 132
BOLT-11 ... 132
BOLT-12 ... 133
BOLTZ ... 133
BORROMEAN RING SIGNATURES ... 133
BOS - BALANCE OF SATOSHIS ... 134
BOSMINER ... 134
BOUNTY ... 134
BPM - BITCOIN POOLED MINING ... 135
BRAIDPOOL ... 135
BRAIINS POOL ... 136
BRAINWALLET ... 136
BRANCH-AND-BOUND ... 137
BRANCHE - BITCOIN ... 137
BRANCHE - GIT ... 137
BRC-20 ... 138
BREEZ ... 138
BROLLUPS ... 138
BSMS ... 139
BSV - BITCOIN SATOSHI VISION ... 139
BTC ... 140
BTCD - BTC SUITE ... 140
BTCMAP ... 140
BTCPAY SERVER ... 141
BTG - BITCOIN GOLD ... 141
BTKN ... 141
BTRUST ... 142
BUCKETING ... 142
BULL BITCOIN ... 142
BULL MARKET ... 143
BULL RUN ... 143
BULLETPROOFS ... 143
BURN ADDRESS ... 143
BUSINESS LOGIC ... 144

C

C++ ... 146
CAHOOTS ... 146
CAKE WALLET ... 146
CANAAN ... 147
CANAL DE PAIMENT ... 147
CANDIDAT - BLOC ... 147
CAPACITÉ DE CANAL LIGHTNING ... 148
CASA ... 148
CASASCIUS ... 148
CASHU ... 149
CBDC ... 149
CBTC - CITREA BITCOIN ... 149
CENSORSHIP RESISTANCE ... 150
CET ... 150
CGMINER ... 150
CHACHA20-POLY1305 ... 151
CHAINALYSIS ... 151
CHAINCODE LABS ... 151

CHAINE EXTERNE 152
CHAINE INTERNE 152
CHAINSPLIT 153
CHAINSTATE/ 153
CHAINWAY LABS 153
CHAINWORK 154
CHAMPION 154
CHANGE 155
CHANGE AVOIDANCE 155
CHANNEL ACCEPTOR 156
CHANNEL ANNOUNCEMENT 156
CHANNEL BREACH 156
CHANNEL.DB 157
CHANNEL FACTORIES 157
CHANNEL JAMMING 158
CHANNEL LEASE 158
CHANNEL POINT 158
CHANNEL RESERVE 159
CHANTOOLS 159
CHARGE UTILE 159
CHAUMIAN COINJOIN 160
CHECKPOINTS 161
CHEMIN DE DÉRIVATION 162
CHEMIN DE RÉCUPÉRATION 163
CHEMIN PRIMAIRE 163
CHIFFRER - CHIFFREMENT 164
CIBLE DE DIFFICULTÉ 164
CIOH 165
CIRCULAR REBALANCE 166
CISA 166
CITREA 167
CLARK - COVENANTLESS ARK 167
CLÉ DE RÉVOCATION 168
CLÉ ÉTENDUE 168
CLÉ MAITRESSE 169
CLÉ PRIVÉE 170
CLÉ PUBLIQUE 170
CLÉ PUBLIQUE COMPRESSÉE 171
CLEARNET 172
CLEMENTINE 172
CLI 173
CLIENT-SIDE VALIDATION 173
CLN - C-LIGHTNING 173
CLN - CORE-LIGHTNING 173
CLONE 174
CLUSTER 174
CLUSTER MEMPOOL 174
CODE DE CHAINE 175
CODE DE CHAINE MAITRE 175
CODE DE PAIMENT RÉUTILISABLE 175
CODEX32 176
COIN CONTROL 176
COINBASE 177
COINJOIN 178
COINJUMBLE 179
COINKITE 180
COINMUX 180
COINOS 180
COINS/ 180
COINSHUFFLE 181
COINSWAP 181
COLD WALLET 182
COLDCARD 182
COLLATÉRAL 183
COLORED COINS 183
COMMERÇANT 183
COMMIT 184
COMMITMENT 184
COMMITMENT FEE 184
COMPACT BLOCK FILTERS 185
COMPACT BLOCK RELAY 185
COMPACTSIZE 186
COMPATIBILITÉ RÉTROSPECTIVE 186
COMPTE 186
CONCATÉNATION 187
CONDENSAT 187
CONFIDENTIAL TRANSACTIONS 188
CONFIRMATION 188
CONNECTOR 188
CONSENSUS 189
CONSENSUS CLEANUP 189
CONSIGNMENT 190
CONSOLIDATION 190
CONTENEUR 191
CONTRACT OPERATION 191
CONTRACT PARTICIPANT 191
CONTRACT RIGHTS 192
CONTRACT STATE 192
CONTRAT INTELLIGENT 192
CONTRAT RGB 192
CONTRIBUTEUR - CORE 193
CONTROL BLOCK 193
CONTROL BOARD 193
COOKIE 194
COOPERATIVE CLOSE 194
COORDINATEUR DE COINJOIN 194
COUNTERPARTY 195
COURBE ELLIPTIQUE 195
COURS LÉGAL 195
COVENANT 196
COVERT ASICBOOST 197
CPFP - CHILD PAY FOR PARENT 197
CPPSRB 197
CPU - CENTRAL PROCESSING UNIT 198
CRYPTANALYSE 198
CRYPTER 198
CRYPTO-ACTIF 198
CRYPTOGRAPHIE 198
CRYPTOLOGIE 198
CRYPTOMONNAIE 199
CTUSD - CITREA USD 199
CURTAILMENT 199

CUSTODY 199
CYPHERPUNKS 200

D

DAEMON 202
DANDELION 202
DARKWALLET 202
DARWINISME MONÉTAIRE 203
DATABASE/ 203
DB.LOG 203
DBC 203
DCA - DOLLAR COST AVERAGING 204
DDOS 204
DEBUG.LOG 204
DÉCENTRALISATION 204
DÉFLATION 205
DEMEURAGE 205
DÉNI PLAUSIBLE 206
DEPEG 206
DÉPÔT 207
DER 207
DÉRIVATION 207
DÉRIVATION ENDURCIE 208
DÉRIVATION NORMALE 208
DGM 209
DIFFICULTÉ 209
DIFFIE-HELLMAN 209
DIFFUSION 210
DIGICASH 210
DIGITAL ARTIFACTS 210
DIRECTED ACYCLIC GRAPH 210
DISTRIBUÉ 211
DKG 211
DLC - DISCREET LOG CONTRACT 211
DLP - DISCRETE LOGARITHM PROBLEM 212
DNS SEEDS 212
DOJO 213
DOS - DENIAL OF SERVICE 213
DOUBLE DÉPENSE 213
DOXXIC CHANGE 214
DRIVECHAIN 215
DUAL FUNDING 215
DUMMY ELEMENT 215
DURESS WALLET 216
DUST 216
DUST LIMIT 216
DUSTING ATTACK 217
DUSTRELAYFEE 217
DYNAFED - DYNAMIC FEDERATIONS 218

E

E-GOLD 220
ECASH - DAVID CHAUM 220
ECASH - XEC 220
ECDH 221
ECDSA 221
ECLAIR 221
ECLIPSE 222
ÉCOLE AUTRICHIENNE 222
EDGE NODE 222
EFFET CANTILLON 223
EFFET DE RÉSEAU 223
ELECTRS 223
ELECTRUM 224
ELECTRUM LIGHTNING 224
ELECTRUM SERVER 224
ELEMENTS 224
ELTOO 225
EMBRANCHEMENT NATUREL 225
EMPREINTE DE PORTEFEUILLE 226
ENDIANNESS 226
ENGRAVING 226
ENTÊTE DE BLOC 227
ENTROPIE 228
ENTROPIE - ANALYSE DE CHAINE 228
ENVOY 229
EPHEMERAL ANCHORS 229
EREBUS 230
ERLAY 230
ESCROW 231
ESMPPS 231
ESPLORA 231
ETF 232
ÉTIQUETAGE 232
ÉVÉNEMENT NOSTR 232
EXIT TRANSACTION 233
EXPLORATEUR DE BLOC 233
EXTENSION BLOCK 234
EXTRA-NONCE 234
EXTRA TRANSACTION PROOF 235

F

F2POOL 238
FARADAY 238
FAUCET 238
FÉDÉRATION 239
FEDIMINT 239
FEE BUMPING 240
FEE_ESTIMATES.DAT 240
FEE RATE - BITCOIN 240
FEE RATE - LIGHTNING 241
FEE SNIPING 241
FEELER CONNECTION 242
FERME DE MINAGE 242
FIAT 243
FIBRE 243
FIDELITY BOND 243
FINALITÉ DE TRANSACTION 244
FINNEY HAL 244
FLAG DAY 245
FLOOD AND LOOT 245
FONCTION DE HACHAGE 245
FONCTION DE HACHAGE TAGUÉE 246
FONCTIONNAIRE 247
FONDATION BITCOIN 247
FONGIBILITÉ 247
FORCE BRUTE 248
FORCE CLOSE 248
FORCED ADDRESS REUSE 249
FORFEIT TRANSACTION 249
FORK 250
FORK - GIT 250
FOSS 251
FOUNDATION DEVICES 251
FOUNDRY USA 251
FOUNTAIN 252
FPGA 252
FPPS - FULL PAY PER SHARE 252
FRAIS DE TRANSACTION 253
FROST 254
FROSTSNAP 254
FULCRUM 255
FULL RBF 255

G

GALOY 258
GAP LIMIT 258
GATEWAY NODE 259
GCS - GOLOMB-CODED SET 259
GENÈSE 259
GENESIS 260
GETBLOCKTEMPLATE 260
GETWORK 260
GHASH.IO 261
GINGER WALLET 261
GIT 261
GITHUB 262
GITLAB 262
GLOBAL STATE 262
GNPA 263
GNU 263
GOLDFINGER 263
GOSSIP 264
GPL 264
GRAFTROOT 264
GRAINE 265
GREEN ADDRESS 266
GREEN ADDRESSES 266
GREEN WALLET 266
GRIEFING ATTACK 266
GROTH16 267
GRPC 267
GUI 267
GUISETTINGS.INI.BAK 268

H

HACHEUR 270
HALVING 270
HARD FORK 270
HARDWARE WALLET 271
HASH160 271
HASH256 272
HASHBOARD 272
HASHCASH 272
HASHRATE 273
HASHRATE HIJACKING 273
HAUTEUR DE BLOC 274
HD - HIERARCHICAL-DETERMINISTIC 274
HEADERS-FIRST SYNC 274
HEURE UNIX 275
HEURISTIQUE D'ANALYSE 275
HEXADÉCIMAL 276

HMAC-SHA512 276
HODL 277
HODL HODL 277
HODL INVOICE 277
HOME MINING 278
HONG-KONG ROUNDTABLE 278
HORODATAGE 279
HOSTED CHANNEL 279
HRP - HUMAN READABLE PART 280
HSM 280
HTLC 281
HTLC ENDORSEMENT 282
HUMAN RIGHTS FOUNDATION 282
HWI 282
HYDRO COOLING 283
HYPER-BITCOINISATION 283
HYPERINFLATION 283

I

I2P 286
IBD - INITIAL BLOCK DOWNLOAD 286
IMMERSION COOLING 286
IMPLÉMENTATION DE BITCOIN 287
INBOUND CAPACITY 287
INDEX - KEY 287
INDEXES/TXINDEX/ 288
INFLATION 288
INPUT 288
INSCRIPTIONS 289
INTERFACE 289
INTERFACE IMPLEMENTATION 289
INVOICE LIGHTNING 290
INVOICE RGB 290
IOU 290
IP_ASN.MAP 290
IP REPORT 291
ISSUE 291

J

JADE 294
JAM 294
JAN3 294
JBOK 295
JIT CHANNEL 295
JIT ROUTING 296
JOININBOX 296
JOINMARKET 296
JOINPOOL 297
JOINSTR 297

K

KEY TELEPORT 300
KEYNÉSIANISME 300
KEYSEND 300
KEYSTORE 301
KNAPSACK SOLVER 301
KRUX 301
KYC - KNOW YOUR CUSTOMER 301

L

L402 304
LABEL 304
LABEL - SILENT PAYMENTS 305
LATENCE 306
LBTC - LIQUID BITCOIN 306
LCB/FT 307
LDK - LIGHTNING DEV KIT 307
LEDGER 307
LEDGER - ENTREPRISE 307
LEVELDB 308
LIANA 308
LIBBITCOIN 308
LIBERTY RESERVE 309
LIBSECP256K1 309
LIBWALLY-CORE 309
LIGHTNING LABS 310
LIGHTNING NETWORK 311
LIGHTNING NETWORK+ 311
LIGHTNING SERVICE PROVIDER 311
LIGHTNING TERMINAL 311

LIGHTSPARK 312
LIMITE D'ÉMISSION 312
LIQUID NETWORK 312
LIQUIDITÉS 313
LIQUIDITY ADS 313
LIQUIDITY SWAP 313
LITD 314
LITECOIN 314
LITTLE-ENDIAN 314
LNBITS 314
LNC - LIGHTNING NODE CONNECT 315
LNCLI 315
LND 315
LNP/BP 316
LNP2PBOT 316
LNURL 316
LOCALBITCOINS 317
LOCK 317
LOGARITHME DISCRET 317
LOI DE GRESHAM 318
LOI DE THIERS 319
LOOP 319
LOOP IN 319
LOOP OUT 320
LOTTERY MINING 320
LUCK 320
LUD 321
LUXOR 321

M

M-OF-N 324
MACAROON 324
MACHANKURA 324
MAGIC NETWORK 325
MAGICAL BITCOIN 325
MAGMA 325
MAINNET 326
MAINTENEUR - CORE 326
MAINTENEUR PRINCIPAL - CORE 326
MAJORITÉ ÉCONOMIQUE 327
MALLÉABILITÉ - TRANSACTION 327
MAPPER 327
MARA 328
MARKET CAP 328
MASF 328
MAST 329
MASTER FINGERPRINT 329
MATT 329
MAX_BLOC_SIZE 330
MAXIMALISTE 330
MÉLANGEUR 330
MEMPOOL 331
MEMPOOL.DAT 331
MEMPOOL.SPACE 331
MERGE 332
MERKLE BLOCK 332
MERKLE SUM TREE 332
MESURE À LA PRISE 332
MÉTADONNÉES 333
MÉTHODE D'ACTIVATION 333
MÉTHODE GÉOMÉTRIQUE 333
MILLIBITCOIN 334
MILLISATOSHI 334
MIMBLEWIMBLE 334
MINAGE 335
MINAGE FUSIONNÉ 336
MINAGE FUSIONNÉ AVEUGLE 336
MINEUR 336
MINIBITS 337
MINING MANAGEMENT SOFTWARE 337
MINING POOL 337
MINISCRIPT 338
MINISKETCH 338
MINITAPSCRIPT 339
MIT DCI 339
MIT X11 340
MITM - MAN-IN-THE-MIDDLE 340
MIXAGE 340
MODÈLE DE SCRIPT 341
MODÈLE DE TRANSACTION 341
MODÈLE TEMPOREL 341
MOSTRO 342
MPP - MULTI-PATH PAYMENTS 342
MS-SMT 343
MTGOX 343
MTP - MEDIAN TIME PAST 343
MULTI PROTOCOL COMMITMENT 344
MULTISIG 344
MUSIG 345
MUSIG-DN 345
MUSIG2 345
MYNODE 346
MYSTERY SHOPPER PAYMENTS 346

N

NAKAMOTO SATOSHI 348
NAMECOIN 348
NAT - NETWORK-ADJUSTED TIME 348
NBITS 348
NERDMINER 349
NESTED SEGWIT 349
NEUTRINO 349
NFT 350
NIP 350
NLOCKTIME 350
NO2X 351
NODL 351
NOEUD 351
NOEUD COMPLET 352
NŒUD DE ROUTAGE 352
NOEUD ÉLAGUÉ 352
NOEUD LIGHTNING 353
NOEUD SPV - NOEUD LÉGER 353
NOISE 354
NONCE 354
NOSTR 355
NPUB 355
NSEC 355
NSEQUENCE 356
NULL DATA 356
NULLDUMMY 356
NUNCHUK 357
NVERSION 357
NWC - NOSTR WALLET CONNECT 357
NYA - NEW YORK AGREEMENT 358

O

OAP - OPEN ASSETS PROTOCOL 360
OBJECTIF 360
OBOE - OFF-BY-ONE ERROR 361
OBSOLÈTE - BLOC 362
OCEAN 362
OCTET 363
OFF-CHAIN 363
OFF-GRID 363
OFFER 364
OMNI 364
ON-CHAIN 364
ON-GRID 364
ONION ADDRESS 365
ONION MESSAGE 365
ONION_PRIVATE_KEY 365
ONION SERVICE 366
ONION_V3_PRIVATE_KEY 366
OOR PAYMENT 366
OP_0 - 0X00 367
OP_0NOTEQUAL - 0X92 367
OP_1 - 0X51 367
OP_1ADD - 0X8B 367
OP_1NEGATE - 0X4F 367
OP_1SUB - 0X8C 367
OP_2 À OP_16 - 0X52 À 0X60 368
OP_2DROP - 0XD6 368
OP_2DUP - 0X6E 368
OP_2OVER - 0X70 369
OP_2ROT - 0X71 369
OP_2SWAP - 0X72 369
OP_3DUP - 0X6F 370
OP_ABS - 0X90 370
OP_ADD - 0X93 370
OP_BOOLAND - 0X9A 371
OP_BOOLOR - 0X9B 371
OP_CAT - 0X7E 371
OP_CHECKHASHVERIFY - CHV 371
OP_CHECKLOCKTIMEVERIFY - 0XB1 371
OP_CHECKMULTISIG - 0XAE 372
OP_CHECKMULTISIGVERIFY - 0XAF 372
OP_CHECKSEQUENCEVERIFY - 0XB2 373
OP_CHECKSIG - 0XAC 373
OP_CHECKSIGADD - 0XBA 373
OP_CHECKSIGFROMSTACK - 0XCC 374
OP_CHECKSIGVERIFY - 0XAD 375
OP_CHECKTEMPLATEVERIFY - 0XB3 375
OP_CODESEPARATOR - 0XAB 375
OP_DEPTH - 0X74 376
OP_DROP - 0X75 376
OP_DUP - 0X76 376
OP_ELSE - 0X67 376
OP_ENDIF - 0X68 376
OP_EQUAL - 0X87 376
OP_EQUALVERIFY - 0X88 377
OP_EVAL 377
OP_FALSE - 0X00 377
OP_FROMALTSTACK - 0X6C 377
OP_GREATERTHAN - 0XA0 377
OP_GREATERTHANOREQUAL - 0XA2 377
OP_HASH160 - 0XA9 378
OP_HASH256 - 0XAA 378
OP_IF - 0X63 378
OP_IFDUP - 0X73 378
OP_INTERNALKEY - 0XCB 378
OP_LESSTHAN - 0X9F 379
OP_LESSTHANOREQUAL - 0XA1 379
OP_MAX - 0XA4 379
OP_MIN - 0XA3 379

OP_NEGATE - 0X8F 379
OP_NIP - 0X77 379
OP_NOP - 0X61 380
OP_NOT - 0X91 380
OP_NOTIF - 0X64 380
OP_NUMEQUAL - 0X9C 380
OP_NUMEQUALVERIFY - 0X9D 381
OP_NUMNOTEQUAL - 0X9E 381
OP_OVER - 0X78 381
OP_PICK - 0X79 381
OP_PUSHDATA1 - 0X4C 381
OP_PUSHDATA2 - 0X4D 382
OP_PUSHDATA4 - 0X4E 382
OP_RETURN - 0X6A 382
OP_RIPEMD160 - 0XA6 382
OP_ROLL - 0X7A 382
OP_ROT - 0X7B 383
OP_SHA1 - 0XA7 383
OP_SHA256 - 0XA8 383
OP_SIZE - 0X82 383
OP_SUB - 0X94 383
OP_SUCCESS 383
OP_SWAP - 0X7C 384
OP_TOALTSTACK - 0X6B 384
OP_TRUE - 0X51 384
OP_TUCK - 0X7D 384
OP_TXHASH - 0XBD 385
OP_VAULT - 0XBB 385
OP_VER - 0X62 386
OP_VERIFY - 0X69 386
OP_WITHIN - 0XA5 386
OPCODES 387
OPENDIME 387
OPENNODE 387
OPENSATS 388
OPENTIMESTAMPS 388
ORACLE 388
ORDINAL NUMBER 388
ORDINALS THEORY 389
ORPHELIN 389
OSINT 390
OUTBOUND CAPACITY 390
OUTPOINT 390
OUTPUT 391
OUTPUT LINKING 391
OUTPUT SCRIPT DESCRIPTORS 392
OVERCLOCKING 393
OVERT ASICBOOST 393
OWNED STATE 394
OWNERSHIP 394
OXT 394

P

P2MS 396
P2P - PAIR-À-PAIR 396
P2P TRANSPORT V2 397
P2PK 397
P2PKH 398
P2POOL 398
P2SH 399
P2SH-P2WPKH 399
P2SH-P2WSH 400
P2TR 400
P2WPKH 401
P2WSH 401
PACKAGE RELAY 402
PAIEMENT ROND 403
PAIEMENT SIMPLE 403
PAIR ENTRANT 404
PAIR SORTANT 404
PAIRING PHRASE 404
PANNE BYZANTINE 405
PAPER WALLET 405
PASSIVE ASSETS 405
PASSPHRASE - BIP-0039 406
PASSPORT 407
PATHFINDING 407
PATOSHI 407
PAY-TO-CONTRACT 408
PAYJOIN 408
PAYMENT HASH 409
PAYNYM 409
PBKDF2 410
PEACH 410
PEDERSEN COMMITMENT 411
PEELING CHAIN 411
PEER DISCOVERY 412
PEERS.DAT 412
PEG-IN 413
PEG-OUT 413
PERCOLATION 413
PÉRIMÉ - BLOC 414
PÉRIODE DE MATURITÉ 414
PHOENIX 415
PHOENIXD 415
PHRASE DE RÉCUPÉRATION 415
PILE 416
PILULE ORANGE 416
PIZZA DAY 417
POINT D'ENTRÉE 417
POINT GÉNÉRATEUR 418
POLAR 418
POLICY - MINISCRIPT 418

POLITIQUE DE MEMPOOL 419
POLITIQUE DE RELAIS 419
POLITIQUE DE RELAIS DES TX 420
POOL DE COINJOIN 420
POOL HOPPING 421
POOL - LIGHTNING 421
PORTAL 421
PORTE DÉROBÉE 421
PORTEFEUILLE 422
PORTEFEUILLE CHAUD 422
POSTMIX 422
POT - PAY ON TARGET 423
PPA 423
PPLNS - PAY PER LAST N SHARES 423
PPS - PAY PER SHARE 424
PPS+ 424
PRAXÉOLOGIE 424
PRÉFÉRENCE TEMPORELLE 425
PRÉFIXES BINAIRES 425
PREIMAGE 426
PREMIUM 426
PREMIX 427
PREUVE DE RÉSERVES 427
PREUVE DE TRAVAIL 428
PRIME DE PROXIMITÉ 429
PRIVATE CHANNEL 429
PROBLÈME DES GÉNÉRAUX BYZANTINS 430
PROBING 430
PROFONDEUR 431
PROOF-OF-KEY DAY 431
PROOF OF STAKE 432
PROP - PROPORTIONAL 432
PROTON WALLET 432
PSAN 433
PSBT 433
PSEUDO-ALÉATOIRE 434
PTLC 434
PUBLIC CHANNEL 435
PUBLIC POOL 435
PULL REQUEST 435
PURGE DE MEMPOOL 436

Q

QR CODE 438
QE - QUANTITATIVE EASING 438
QUBIT 439

R

RACE ATTACK 442
RACINE DE MERKLE 442
RANGEPROOFS 442
RAPID GOSSIP SYNC 443
RASPIBLITZ 443
RAW TRANSACTION 443
RBF - REPLACE-BY-FEE 444
RÉCOMPENSE DE BLOC 444
RÉCURSIF - COVENANT 445
REDEEM 445
REDEEMSCRIPT 446
RÈGLES DE CONSENSUS 446
RÈGLES DE STANDARDISATION 447
REGTEST 448
RELAIS 448
RELAIS NOSTR 448
RENDEZ-VOUS ROUTING 449
REPLACEMENT CYCLING 449
REPLAY ATTACK 450
RÉSEAU BITCOIN 450
RÉSERVES FRACTIONNAIRES 451
RÉSISTANCE AU PARTITIONNEMENT 451
RESYNCHRONISATION 451
RÉTENTION DE BLOC 452
RÉUTILISATION D'ADRESSE 453
RÉUTILISATION D'ADRESSE - EXTERNE 453
RÉUTILISATION D'ADRESSE - INTERNE 454
REVAULT 454
REVOKEANDACK 455
RGB 455
RICOCHET 456
RIOT PLATFORMS 456
RIPEMD160 457
RISC ZERO 457
RISQUE DE CONTREPARTIE 457
ROAST 458
ROBOSATS 458
ROLLUP 459
ROUND ARK 459
ROUTAGE EN OIGNON 460
RPC - REMOTE PROCEDURE CALL 460
RPOW 460
RSK 461
RSMPPS 461
RTL - RIDE THE LIGHTNING 462
RUNES 462
RUST BITCOIN 462
RUST-LIGHTNING 463

S

STRATEGIC BITCOIN RESERVE 466
SAMOURAI WALLET 466
SAT - SATOSHI 466
SATOCHIP 466
SATODIME 467
SATOSHIDICE 467
SATOSHILABS 467
SATSCARD 467
SBI CRYPTO 468
SCALABILITÉ 468
SCB - STATIC CHANNEL BACKUP 468
SCHEMA 469
SCHNORR 469
SCID - SHORT CHANNEL ID 469
SCORE DE BOLTZMANN 470
SCORE BASED METHOD 470
SCRIPT 470
SCRIPTLESS SCRIPTS 471
SCRIPTPUBKEY 471
SCRIPTSIG 472
SCRIPTWITNESS 473
SDK - SOFTWARE DEVELOPMENT KIT 473
SE - SPARK ENTITY 473
SEAL DEFINITION 474
SECP256K1 474
SECP256R1 475
SECPOOL 475
SECURE ELEMENT 475
SEED NODES 476
SEEDKEEPER 476
SEEDQR 477
SEEDSIGNER 477
SEEDXOR 478
SEGWIT 478
SEGWIT V0 479
SEGWIT V1 479
SEGWIT2X 479
SEIGNEURIAGE 480
SÉLECTION DES PIÈCES 480
SELF-CUSTODY 480
SELFISH MINING 481
SENTINEL 481
SÉQUENCEUR 482
SETTINGS.JSON 482
SHA256 482
SHA512 483
SHAMIR SECRET SHARING - SSS 483
SHARDS - LIGHTNING 483
SHARDS - RGB 484
SHAREDCOIN 484
SHARES 484
SHARES DIFFICULTY 485
SHITCOIN 486
SHOR 486
SIDECAR CHANNEL 486
SIDECAR TICKET 487
SIDECHAIN 487
SIGHASH_ALL 488
SIGHASH_ALL | SIGHASH_ACP 488
SIGHASH_ANYPREVOUT 488
SIGHASH_ANYPREVOUTANYSCRIPT 489
SIGHASH FLAG 489
SIGHASH_NONE - 0X02 490
SIGHASH_NONE | SIGHASH_ACP 490
SIGHASH_SINGLE - 0X03 490
SIGHASH_SINGLE | SIGHASH_ACP 490
SIGN-TO-CONTRACT 491
SIGNALING 491
SIGNATURE AVEUGLE 491
SIGNATURE NUMÉRIQUE 492
SIGNER À L'AVEUGLE 492
SIGNET 493
SIGOPS 493
SILENT PAYMENTS 494
SILK ROAD 495
SIMPLICITY 496
SIMPLIFIED PAYMENT VERIFICATION 497
SINGLE-SIG 497
SINGLE-USE SEAL 497
SLIP 498
SMALL BLOCKERS 498
SMPPS 498
SMT - SPARSE MERKLE TREE 499
SNAPSHOT 499
SNARK 499
SO - SPARK OPERATOR 500
SOFT FORK 500
SOLO MINING 500
SOMME DE CONTRÔLE 501
SOROBAN 501
SORTIE LA PLUS GRANDE 502
SORTIE NON RENTABLE 502
SOURCEFORGE 502
SPARK 503
SPARK SERVICE PROVIDER 503
SPARK TREE 503
SPARKSCAN 504
SPARROW WALLET 504
SPECTER 504
SPEEDY TRIAL 505
SPHINX 505
SPIDERPOOL 505
SPIRAL 506
SPLICING 506
SPOF 507
SPREAD - WST 507
SPY MINING 508
SRC-20 508

STABLECOIN 508
STACKER 508
STALE SHARE 509
STAMPS 509
STARK 510
START9 510
STASH 510
STATE EXTENSION 511
STATE TRANSITION 511
STATECHAIN 511
STEALTH ADDRESS 512
STÉGANOGRAPHIE 512
STONEWALL 512
STONEWALL X2 513
STRATUM 514
STRATUM V2 515
STUCKLESS PAYMENT 515
SUBMARINE SWAP 516
SUBVENTION DE BLOC 516
SURCOUCHE 516
SURFACE D'ATTAQUE 517
SWAP-IN POTENTIAM 517
SWAPROOT 518
SWEEP TRANSACTION 518
SWISS BITCOIN PAY 519
SYBIL 519
SYNONYM 519

T

TAIL EMISSION 522
TAPD 522
TAPROOT 523
TAPROOT ASSETS PROTOCOL 523
TAPROOT CHANNEL 524
TAPSCRIPT 524
TAPSIGNER 524
TARO 524
TCP 525
TÉMOIN DE TRANSACTION 525
TEN31 525
TERMINAL CONSIGNEMENT 525
TESTNET 526
TESTNET RESET 526
TESTNET4 526
THE DAO 527
THE EYE OF SATOSHI 527
THUNDERHUB 527
TIDES 528
TIME-DILATION ATTACK 528
TIME WARP 529
TIMELOCK 530
TLV 530
TOR 530
TPRV 531
TPUB 531
TRAMPOLINE ROUTING 531
TRANSACTION COLLABORATIVE 532
TRANSACTION D'ENGAGEMENT 532
TRANSACTION DE FINANCEMENT 532
TRANSACTION DE NOTIFICATION 533
TRANSACTION DE PÉNALITÉ 533
TRANSACTION NON CONFIRMÉE 534
TRANSACTION PINNING 534
TRANSACTION STANDARD 534
TRANSITION BUNDLE 535
TREZOR 535
TRIMMED HTLC 536
TUMBLEBIT 536
TURBO CHANNEL 537
TWEAK 537
TWENTYTWO DEVICES 538
TX - TRANSACTION 538
TX0 539
TXID - TRANSACTION IDENTIFIER 539
TYPE DE DEVISE 539

U

UASF 542
UDP 542
UMBREL 542
UNIQUE ASSETS 543
UNIVERSE 543
UNIX 543
UPRV 543
UPUB 543
URI 544
UTREEXO 544
UTXO 545
UTXO SET 546

V

VALENCY 548
VALIDATIONLESS MINING 548
VALUE OVERFLOW INCIDENT 548
VANITY ADDRESS 549
VANITYGEN 549
VARIANCE 549
VAULT 550
VBYTES 550
VERSIONNAGE 551
VEXL 551
VIABTC 551
VIN 552
VIRGIN BITCOIN 552
VLS 552
VOLATILITÉ 553
VOLTAGE 553
VOUT 553
VPRV 554
VPUB 554
VTXO 554
VULNERABILITE DE MURCH-ZAWY 554

W

WABISABI 558
WALLET.DAT 558
WIF - WALLET IMPORT FORMAT 558
WALLET OF SATOSHI 558
WALLETS/DB.LOG 559
WASABI WALLET 559
WATCH-ONLY WALLET 559
WATCHMEN 559
WATCHTOWER 560
WEIGHT UNIT - WU 560
WHALE 561
WHIRLPOOL 561
WHIRLPOOL STAT TOOL 562
WHITE PAPER 562
WITNESS TRANSACTION 562
WITNESSSCRIPT 562
WIZARDSARDINE 563
WRENCH ATTACK 563
WTXID 564
WUMBO CHANNEL 564

X

X-ONLY PUBLIC KEYS 566
XBT 566
XMR - MONERO 567
XOR 567
XPRV 568
XPUB 568

Y

YPRV 570
YPUB 570

Z

ZAP 572
ZBD 572
ZEROCONF 572
ZEROLINK 573
ZEROSYNC 573
ZEUS 573
ZKP - ZERO-KNOWLEDGE PROOF 573
ZKSNACKS 574
ZMQ - ZEROMQ 574
ZOMBIE CHANNEL 574
ZPRV 574
ZPUB 574

A

ACINQ

ORGANISATION

Entreprise française spécialisée dans le développement de solutions pour le Lightning Network. ACINQ est à l'origine d'Eclair, une implémentation majeure du protocole Lightning écrite en Scala. L'entreprise développe également Phoenix, un portefeuille mobile *self-custodial* qui intègre un nœud Lightning directement sur le téléphone de l'utilisateur. ACINQ exploite par ailleurs un des plus grands nœuds Lightning du réseau.

Termes associés : **LIGHTNING NETWORK** | **PHOENIX**

ACTIF AU PORTEUR

ÉCONOMIE ET RÉGULATION | EN : BEARER ASSET

Bien ou instrument financier dont la propriété est déterminée uniquement par la possession physique, sans nécessiter l'intervention d'un registre centralisé, d'un intermédiaire ou d'une inscription nominative. Celui qui détient l'actif en est le propriétaire légitime. Les billets de banque, les pièces d'or et les obligations au porteur sont des exemples d'actifs au porteur.

Ce type d'actif offre l'avantage de ne dépendre d'aucun tiers pour valider ou autoriser un transfert de propriété. Il confère une souveraineté directe à son détenteur, mais implique aussi une responsabilité : la perte ou le vol entraîne la perte définitive de l'actif.

Le bitcoin en *self-custody* est un actif numérique au porteur. La possession des clés privées qui sécurisent des bitcoins confère un contrôle exclusif sur ces fonds, sans qu'aucune banque, institution ou registre centralisé n'intervienne.

Termes associés : **CLÉ PRIVÉE** | **FIAT** | **SELF-CUSTODY**

ADAPTOR SIGNATURE

CRYPTOGRAPHIE

Méthode cryptographique permettant de combiner une vraie signature avec une signature supplémentaire (appelée « adaptor signature ») pour révéler une donnée secrète. Cette méthode fonctionne telle que la connaissance de deux éléments parmi la signature valide, l'adaptor signature et le secret permet de déduire le troisième manquant. Une des propriétés intéressantes de cette méthode est que si nous connaissons l'adaptor signature de notre pair et le point spécifique sur la courbe elliptique lié au secret utilisé pour calculer cette adaptor signature, nous pouvons alors dériver notre propre adaptor signature qui correspondra avec le même secret, et ce, sans jamais avoir accédé directement au secret lui-même. Dans un échange entre deux parties prenantes ne se faisant pas confiance, cette technique permet un dévoilement simultané de deux informations sensibles entre les participants. Ce processus élimine la nécessité de confiance lors de transactions instantanées telles qu'un coinswap ou un Atomic Swap. Prenons un exemple pour bien comprendre. Alice et Bob souhaitent s'envoyer 1 BTC chacun, mais ils ne se font pas confiance. Ils vont donc utiliser des adaptors signatures pour annihiler le besoin de confiance envers l'autre partie dans cet échange (c'est donc un échange « atomique »). Ils

procèdent comme ceci :

- Alice lance cet échange atomique. Elle crée une transaction m_A qui envoie 1 BTC vers Bob. Elle crée une signature s_A qui permet de valider cette transaction grâce à sa clé privée p_A ($P_A = p_A \cdot G$), et en utilisant un nonce n_A et un secret t ($N_A = n_A \cdot G$ et $T = t \cdot G$) :

$$s_A = n_A + t + H(N_A + T \parallel P_A \parallel m_A) \cdot p_A$$

- Alice calcule l'adaptor signature s'_A à partir du secret t et de sa vraie signature s_A :

$$s'_A = s_A - t$$

- Alice envoie à Bob son adaptor signature s'_A, sa transaction non signée m_A, le point correspondant au secret T et le point correspondant au nonce N_A. Nous appelons ces informations un « adaptor ». Notons qu'avec simplement ces informations, Bob n'est pas en capacité de récupérer le BTC d'Alice.
- En revanche, Bob peut vérifier qu'Alice n'est pas en train de l'entourlouper. Pour ce faire, il vérifie que l'adaptor signature d'Alice s'_A correspond bien à la transaction promise m_A. Si l'équation suivante est juste, alors il est persuadé que l'adaptor signature d'Alice est valide :

$$s'_A \cdot G = N_A + H(N_A + T \parallel P_A \parallel m_A) \cdot P_A$$

- Cette vérification donne à Bob des garanties de la part d'Alice, de telle sorte qu'il peut continuer le processus d'échange atomique sereinement. Il va alors créer à son tour sa propre transaction m_B envoyant 1 BTC à Alice et sa propre adaptor signature s'_B qui sera liée avec le même secret t que seule Alice connait pour le moment (Bob n'a pas connaissance de cette valeur t, mais uniquement de son point correspondant T qu'Alice lui a fourni) :

$$s'_B = n_B + H(N_B + T \parallel P_B \parallel m_B) \cdot p_B$$

- Bob envoie à Alice son adaptor signature s'_B, sa transaction non signée m_B, le point correspondant au secret T et le point correspondant au nonce N_B. Alice peut désormais combiner l'adaptor signature de Bob s'_B avec le secret t, dont elle seule a connaissance, afin de calculer une signature valide s_B pour la transaction m_B qui lui envoie le BTC de Bob :

$$s_B = s'_B + t$$
$$(s'_B + t) \cdot G = N_B + T + H(N_B + T \parallel P_B \parallel m_B) \cdot P_B$$

- Alice diffuse cette transaction m_B signée sur la blockchain Bitcoin afin de récupérer le BTC que Bob lui a promis. Bob prend connaissance de cette transaction sur la blockchain. Il est donc en capacité d'en extraire la signature $s_B = s'_B + t$. À partir de cette information, Bob peut isoler le fameux secret t dont il avait besoin :

$$t = (s'_B + t) - s'_B = s_B - s'_B$$

- Or, ce secret t était la seule information manquante à Bob pour produire la signature valide s_A, à partir de l'adaptor signature d'Alice s'_A, qui lui permettra de valider la transaction m_A qui envoie un BTC depuis Alice vers Bob. Il calcule alors s_A et diffuse à son tour la transaction m_A :

$$s_A = s'_A + t$$

$$(s'_A + t) \cdot G = N_A + T + H(N_A + T \parallel P_A \parallel m_A) \cdot P_A$$

Termes associés : **ATOMIC SWAP** | **COINSWAP**

ADDR

RÉSEAU

Message réseau anciennement utilisé sur Bitcoin pour communiquer les adresses des nœuds qui acceptent des connexions entrantes. Cet ancien format, se limitant à 128 bits par adresse, était seulement adapté aux adresses IPv6, IPv4 et aux adresses Tor de version 2. Face à l'arrivée de nouveaux protocoles comme Tor V3 et la nécessité de disposer d'une meilleure évolutivité pour de futurs protocoles réseau, le format `addr` a été supplanté par `addrv2`, introduit dans le BIP-0155.

Termes associés : **ADDRV2** | **BIP-0155**

ADDR.DAT

RÉSEAU

Nom de l'ancien fichier utilisé dans Bitcoin Core pour stocker des informations sur les pairs (c'est-à-dire, les nœuds) du réseau avec lesquels le nœud de l'utilisateur a interagi ou peut potentiellement interagir. Ce fichier a été remplacé par le fichier `peers.dat` depuis la version 0.7.0.

Termes associés : **BITCOIN CORE** | **PEERS.DAT**

ADDRESS SPOOFING

ATTAQUE | FR : USURPATION D'ADRESSE

Attaque où un acteur malveillant crée une adresse (ou tout autre identifiant de paiement) ressemblant fortement à celle de la victime. Le but est de tromper l'utilisateur en l'amenant à copier cette mauvaise adresse lors d'une transaction, ce qui conduit à l'envoi des bitcoins à l'attaquant au lieu de la destination prévue.

L'attaquant exploite la précipitation de l'utilisateur qui peut copier la mauvaise adresse s'il réalise sa transaction sans vérifier avec précision son exactitude. En général, pour mettre en place cette attaque, l'attaquant effectue des paiements avec de petites sommes vers le portefeuille de la victime pour intégrer la fausse adresse dans son historique de transactions. Cette attaque est plutôt utilisée avec les altcoins, où il est courant de réutiliser les mêmes adresses de réception, contrairement à Bitcoin où l'utilisation d'adresses vierges pour chaque transaction est une pratique plus répandue. Cependant, les utilisateurs de Bitcoin ne sont pas à l'abri de cette attaque.

Une autre méthode pour mettre la mauvaise adresse devant la victime est l'utilisation de logiciels de gestion de portefeuille frauduleux qui imitent des logiciels

légitimes, ou la modification de l'adresse lorsqu'une machine est compromise, entre le moment où elle est copiée et celui où la transaction est construite. On parle alors parfois d'« *address swapping* ».

Pour se protéger contre ces différentes méthodes d'attaque, il est important de vérifier plusieurs caractères de l'adresse, surtout au niveau de sa *checksum* (à la fin), sur l'écran du périphérique de signature avant de signer la transaction.

On parle également parfois d'Address Poisoning pour désigner cette attaque.

Termes associés : **ADRESSE DE RÉCEPTION** | **SOMME DE CONTRÔLE**

ADDRMAN

RÉSEAU

Module de Bitcoin Core qui gère les adresses des pairs connus sur le réseau. Il maintient deux tables en mémoire : la table « new », qui contient les adresses découvertes mais pas encore testées (1024 seaux de 64 entrées), et la table « tried », qui contient les adresses auxquelles le nœud s'est déjà connecté avec succès (256 seaux de 64 entrées). Ces tables sont sérialisées dans le fichier `peers.dat`.

Le module a été introduit par Pieter Wuille en 2012, en remplacement du système précédent qui stockait les adresses dans `addr.dat` sans structure de bucketing. Son fonctionnement a été renforcé par l'ajout des *feeler connections* et du mécanisme *test-before-evict*, qui permettent de vérifier la joignabilité des adresses avant de les promouvoir ou de les évincer.

Termes associés : **ECLIPSE** | **PEERS.DAT**

ADDRV2

RÉSEAU

Évolution proposée avec le BIP-0155 du message `addr` sur le réseau Bitcoin. Le message `addr` servait à diffuser les adresses de nœuds qui acceptent des connexions entrantes, mais il était limité à des adresses de 128 bits. Cette taille était adéquate pour les adresses IPv6, IPv4, et Tor V2, mais insuffisante pour d'autres protocoles. La version mise à jour `addrv2` est conçue pour supporter des adresses plus longues, notamment les services cachés Tor V3 de 256 bits, ainsi que d'autres protocoles réseau tels que I2P ou de futurs protocoles.

Termes associés : **ADDR** | **BIP-0155**

ADRESSE DE RÉCEPTION

PORTEFEUILLE | EN : BITCOIN ADDRESS

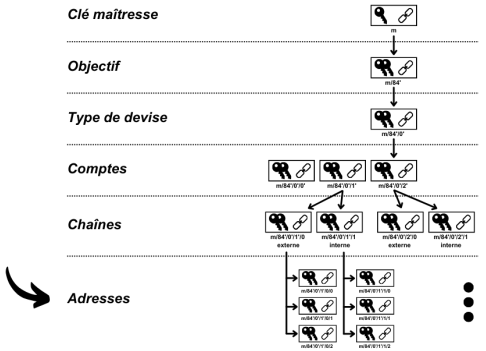
Information utilisée pour recevoir des bitcoins. Une adresse est généralement construite en hachant une clé publique, à l'aide de `SHA256` et de `RIPEMD160`, et en ajoutant des métadonnées à ce condensat. Les clés publiques utilisées pour construire une adresse de réception font partie du portefeuille de l'utilisateur et sont donc dérivées depuis sa graine. Par exemple, les adresses SegWit sont composées des informations suivantes :

- Un HRP pour désigner « bitcoin » : `bc` ;
- Un séparateur : `1` ;

- La version de SegWit utilisée : `q` ou `p` ;
- La charge utile : le condensat de la clé publique (ou directement la clé publique dans le cas de Taproot) ;
- La somme de contrôle : un code BCH.

Mais une adresse de réception peut également représenter autre chose en fonction du modèle de script utilisé. Par exemple, les adresses P2SH sont construites à l'aide du hachage du script. Les adresses Taproot, elles, contiennent directement la clé publique tordue (*tweaked*) sans qu'elle soit hachée.

Une adresse de réception peut être représentée sous la forme d'une chaîne de caractères alphanumériques ou sous la forme d'un QR code. Chaque adresse peut être utilisée plusieurs fois, mais c'est une pratique très déconseillée. En effet, dans le but de maintenir un certain niveau de confidentialité, il est conseillé de n'utiliser chaque adresse Bitcoin qu'une seule fois. Il faut en générer une nouvelle pour tout paiement entrant vers son portefeuille. Une adresse est encodée en `Bech32` pour les adresses SegWit V0, en `Bech32m` pour les adresses SegWit V1, et en `Base58check` pour les adresses Legacy. D'un point de vue technique, recevoir du bitcoin se traduit par le fait de posséder la clef privée associée à une clef publique (et donc à une adresse). Lorsque quelqu'un reçoit des bitcoins, l'émetteur met à jour la contrainte existante sur leur dépense afin que seul le récipiendaire puisse désormais avoir ce pouvoir.



Termes associés : **CLÉ PUBLIQUE** **TAPROOT**

ADRESSE STATIQUE

CONFIDENTIALITÉ | EN : STATIC ADDRESS

Dans le cadre des Silent Payments, désigne un identifiant unique qui permet de recevoir des paiements sans pour autant produire de réutilisation d'adresse, sans interaction et sans lien visible on-chain entre les différents paiements et l'adresse statique. Cette technique élimine le besoin de générer de nouvelles adresses de réception vierges pour chaque transaction, ce qui permet d'éviter les interactions habituelles dans Bitcoin où le destinataire doit fournir une nouvelle adresse au payeur. C'est un peu l'équivalent du code de paiement réutilisable dans le cadre du BIP-0047.

Cette adresse est composée de deux clés publiques : B_{scan} pour le scan et B_{spend} pour la dépense, concaténées pour former l'adresse statique $B = B_{scan} \parallel B_{spend}$. Le destinataire publie cette adresse, permettant aux expéditeurs de dériver des adresses de paiement uniques sans interaction supplémentaire avec le destinataire. Pour gérer plusieurs sources de paiements distinctes, on peut ajouter un label à B_{spend}, créant ainsi plusieurs adresses statiques labellisées (à partir de B_1, B_2, etc.). Cela permet de ségréguer les paiements tout en utilisant une seule adresse de base, réduisant ainsi la charge de travail pour le scan de la blockchain. Toutefois, toutes les adresses statiques d'une entité peuvent être facilement associées en raison de l'utilisation commune de B_{scan}.

Terme associé : **SILENT PAYMENTS**

AEZEED

LIGHTNING NETWORK

Format de phrase de récupération spécifique à LND, utilisé pour générer et sauvegarder les clés privées du portefeuille *onchain* associé à un nœud Lightning. Comme le BIP-0039, l'aezeed produit une phrase mnémonique de 24 mots, mais corrige deux limitations majeures de ce standard : l'absence de versionnage et l'absence de date de création du portefeuille.

La base d'une phrase aezeed mesure 19 octets avec : 1 octet de version (permettant au portefeuille de savoir comment re-dériver les adresses), 2 octets d'horodatage exprimé en jours depuis le bloc genesis de Bitcoin (ce qui couvre les dates jusqu'en 2188), et 16 octets d'entropie servant à dériver la *seed*. L'horodatage indique au logiciel à partir de quand scanner la blockchain.

Ces 19 octets sont chiffrés avec AEZ, un schéma AEAD résistant au réemploi de *nonce*, à l'aide d'une clé dérivée par Scrypt (`n=32768`, `r=8`, `p=1`) depuis une passphrase optionnelle (par défaut : la chaîne « aezeed »). Le texte chiffré de 33 octets est encodé en 24 mots via la même liste que le BIP-0039. Contrairement au BIP-0039, la phrase mnémonique est elle-même un texte chiffré et il est possible de changer la passphrase ou de mettre à jour les paramètres de chiffrement sans générer de nouvelles clés.

Termes associés : **BIP-0039** | **LND**

AGORISME

ÉCONOMIE ET RÉGULATION	EN : AGORISM

Philosophie politique fondée par Samuel Edward Konkin III dans les années 1970. Elle est une extension du libertarianisme, mettant en avant une action directe pour s'extraire des jougs de l'autorité étatique par le biais de la contre-économie, une pratique économique qui se déroule en dehors des cadres régulés par le gouvernement. L'idéologie agoriste repose sur le jusnaturalisme, qui affirme que les droits naturels des individus sont supérieurs aux lois imposées par l'État. Cela inclut la primauté de la propriété privée, le respect de l'intégrité physique, et la liberté de contracter. Les agoristes rejettent toute forme de participation politique traditionnelle comme le vote, qu'ils considèrent comme une validation de l'autorité coercitive de l'État. Ils aspirent à une société où les échanges économiques et sociaux se déroulent librement dans un marché ouvert, appelé l'Agora, visant ainsi à une révolution pacifique pour éroder progressivement le pouvoir de l'État. Leur emblème est d'ailleurs « A3 », pour « Agora, Anarchie, Action ». Le principe de l'agorisme est décrit dans le *Manifeste néo-Libertarien* paru en 1980. Ce papier est établi sur le *Manifeste Libertarien* de Murray Rothbard, mais il va encore plus loin. Beaucoup de bitcoiners se réclament de l'agorisme et pensent que Bitcoin en est l'outil parfait.

Terme associé : **CYPHERPUNKS**

AIR COOLING

MINAGE	FR : REFROIDISSEMENT À L'AIR

Système de refroidissement utilisé pour les ASICs dans le cadre du minage de Bitcoin. L'*air cooling* consiste à utiliser des ventilateurs pour dissiper la chaleur générée par les composants. L'air ambiant est aspiré pour refroidir les puces avant d'être expulsé. Cette méthode de refroidissement est la plus répandue, car les ventilateurs sont présents directement sur les machines à l'achat. C'est généralement la moins onéreuse à l'acquisition, mais elle est moins efficace comparée à des techniques telles que l'*hydro cooling* ou l'*immersion cooling*.

Termes associés : **HYDRO COOLING** | **IMMERSION COOLING**

AIR-GAPPED

PORTEFEUILLE	FR : ISOLÉ DU RÉSEAU

Qualifie un dispositif informatique qui est physiquement isolé de tout réseau, y compris Internet, le Wi-Fi, le Bluetooth ou toute autre connexion filaire ou sans fil. Dans le contexte de Bitcoin, un appareil air-gapped est utilisé pour sécuriser les clés privées en les gardant à l'écart de tout vecteur d'attaque en ligne.

Le principe est simple : si un appareil n'est jamais connecté à un réseau, il devient extrêmement difficile pour un attaquant d'y accéder à distance pour extraire des informations sensibles. Un *hardware wallet* air-gapped signe les transactions hors ligne. Les données sont échangées avec l'ordinateur connecté à Internet par un moyen physique, typiquement via une carte microSD ou un QR code, sans jamais établir de connexion directe.

Cette approche représente un des niveaux de sécurité les plus élevés pour

la conservation de bitcoins. Des appareils comme le Coldcard, le Passport de Foundation Devices ou le Jade de Blockstream proposent ce mode de fonctionnement.

Termes associés : **ADRESSE DE RÉCEPTION** | **COLD WALLET**

AJUSTEMENT DE LA DIFFICULTE

PROTOCOLE | EN : DIFFICULTY ADJUSTMENT

L'ajustement de la difficulté est un processus périodique qui redéfinit la cible de difficulté pour le mécanisme de la preuve de travail (le minage) sur Bitcoin. Cet évènement intervient tous les 2016 blocs (environ toutes les deux semaines). Il vient augmenter ou baisser le facteur de difficulté (également nommé la cible de difficulté), en fonction de la rapidité à laquelle les 2016 derniers blocs ont été trouvés. L'ajustement vise à conserver un taux de production de blocs stable et prévisible, à une fréquence d'un bloc toutes les 10 minutes, malgré les variations de la puissance de calcul déployée par les mineurs. La modification de la difficulté lors de l'ajustement est limitée à un facteur 4. La formule exécutée par les nœuds pour calculer la nouvelle cible est la suivante :

$$N = A \cdot \left(\frac{T}{1\,209\,600} \right)$$

Où :

- N : la nouvelle cible ;
- A : l'ancienne cible ;
- T : le temps écoulé entre le premier et le dernier bloc de la période en secondes ;
- $1\,209\,600$: le temps cible en secondes pour 2016 blocs à 10 minutes d'intervalle (2016×600).

En raison d'un bug off-by-one introduit par Satoshi Nakamoto, T mesure en réalité la durée de 2015 intervalles de blocs, et non 2016. Le calcul compare en effet l'horodatage du premier bloc de la période (bloc n) à celui du dernier (bloc $n + 2015$), ce qui ne couvre que 2015 intervalles. Un calcul correct aurait comparé le dernier bloc de la période précédente (bloc $n - 1$) au dernier bloc de la période actuelle (bloc $n + 2015$), soit 2016 intervalles. Ce bug introduit un biais d'environ 0,05 % vers une difficulté légèrement trop élevée par rapport à ce qu'elle devrait être. Il est surtout à l'origine de la faisabilité de l'attaque *time warp*, car les périodes de retarget ne se chevauchent pas.

En français, on parle également parfois de « reciblage » pour évoquer l'ajustement.

Termes associés : **CIBLE DE DIFFICULTÉ** | **PREUVE DE TRAVAIL**

ALBY

ORGANISATION

Entreprise spécialisée dans les solutions de paiement Lightning. Alby propose un écosystème de produits complémentaires : *Alby Hub, un nœud Lightning* self-custodial* qui peut être auto-hébergé, déployé dans le cloud ou utilisé

comme interface d'un nœud classique ;

- l'extension de navigateur Alby, qui permet d'effectuer des paiements Lightning et de s'authentifier sur des sites web directement depuis le navigateur ;
- Alby Go, une application mobile pour gérer ses paiements en déplacement.

Alby fournit également des API et le protocole *Nostr Wallet Connect* (NWC) pour permettre aux développeurs d'intégrer les paiements Lightning dans des applications tierces.

Terme associé : **LIGHTNING NETWORK**

ALBY HUB

LIGHTNING NETWORK

Application open-source qui permet de faire fonctionner un nœud Lightning Network personnel et de le connecter à diverses applications via le protocole Nostr Wallet Connect (NWC). Alby Hub s'installe sur un serveur personnel, un ordinateur ou dans le cloud, et sert de centre de contrôle pour les paiements Lightning de l'utilisateur. Alby Hub est non-custodial : l'utilisateur conserve ses clés privées et le contrôle total de ses fonds. Il utilise une implémentation LDK (*Lightning Development Kit*) pour la gestion du nœud Lightning.

Termes associés : **LIGHTNING NETWORK** **NOSTR**

ALERT SYSTEM

HISTOIRE | FR : SYSTÈME D'ALERTE

Mécanisme historique de Bitcoin qui permettait de diffuser des messages d'urgence à l'ensemble des nœuds du réseau. Introduit par Satoshi Nakamoto dans la version v0.3.10, peu après le *value overflow incident* d'août 2010, ce système servait à avertir les opérateurs de nœuds en cas de bug critique ou de problème réseau.

Les alertes étaient relayées via le réseau pair-à-pair de Bitcoin, de la même manière que les transactions et les blocs. Seuls les messages signés avec une clé privée ECDSA spécifique étaient considérés comme valides. Cette clé était initialement détenue par Satoshi seul, puis partagée avec quelques développeurs comme Gavin Andresen. Dans les premières versions (jusqu'à v0.3.20), une alerte valide déclenchait un « *safe mode* » qui désactivait automatiquement les commandes RPC liées aux transactions, ce qui donnait à ses détenteurs un pouvoir considérable sur le réseau.

En raison du risque de centralisation et de la possibilité d'usages abusifs, le système a été progressivement abandonné : le code a été retiré dans Bitcoin Core v0.13.0, et une alerte finale « Alert Key Compromised » a été diffusée pour invalider définitivement tout futur message. La clé privée a ensuite été rendue publique en 2018.

Termes associés : **CHECKPOINTS** **VALUE OVERFLOW INCIDENT**

ALGORITHME

‹/› INFORMATIQUE | EN : ALGORITHM

Suite finie et non ambiguë d'instructions permettant de réaliser une tâche. Dans le cadre de l'informatique, il s'agit d'un processus écrit dans un langage de programmation qui indique à un ordinateur comment effectuer une mission.

Terme associé : **FONCTION DE HACHAGE**

ALTCOIN

Ⓢ ÉCONOMIE ET RÉGULATION

Désigne toute cryptomonnaie autre que le bitcoin (BTC). Le terme « altcoin » est la contraction de « alternative » et de « coin » (pièce alternative). Certains bitcoiners maximalistes parlent également de « shitcoins » pour désigner les altcoins.

Terme associé : **SHITCOIN**

ALUVM

RGB

Sigle de « *Algorithmic Logic Unit Virtual Machine* ». Machine virtuelle à registres spécifiquement conçue pour l'exécution déterministe de tâches de calcul distribuées et la validation de *smart contracts*, notamment dans le cadre des contrats RGB. AluVM a initialement été développé par Dr Maxim Orlovsky et est actuellement maintenu par l'association LNP/BP Standards.

Termes associés : **CONTRAT INTELLIGENT** | **RGB**

AMBOSS

✱ OUTIL

Amboss est une plateforme d'analyse et de monitoring du Lightning Network, fondée en 2021 par Jesse Shrader et Anthony Potdevin. Elle fournit des données détaillées sur la topologie du réseau, les nœuds, les canaux et les flux de liquidité, permettant aux opérateurs de nœuds d'optimiser leur gestion. Amboss propose également Magma, une place de marché dédiée à la liquidité Lightning, où les opérateurs peuvent acheter et vendre de la capacité de canal.

Termes associés : **CANAL DE PAIMENT** | **LIGHTNING NETWORK**

AML - ANTI-MONEY LAUNDERING

Ⓢ ÉCONOMIE ET RÉGULATION | FR : LUTTE CONTRE LE BLANCHIMENT D'ARGENT

Désigne l'ensemble des procédures, lois et régulations destinées à prévenir le blanchiment d'argent. Ces règles obligent les institutions financières comme les plateformes d'échange de bitcoins à surveiller activement les transactions de leurs clients, à effectuer des vérifications d'identité, à tenir des registres et à signaler les activités suspectes aux autorités.

Terme associé : **KYC - KNOW YOUR CUSTOMER**

ANALYSE DE CHAINE

CONFIDENTIALITÉ | EN : CHAIN ANALYSIS

Pratique qui regroupe toutes les méthodes permettant de tracer les flux de bitcoins sur la blockchain. De façon générale, l'analyse de chaîne s'appuie sur l'observation de caractéristiques sur des échantillons de transactions antérieures. Elle consiste ensuite à repérer ces mêmes caractéristiques sur une transaction que l'on souhaite analyser, et à en déduire des interprétations vraisemblables. Cette méthode de résolution de problème à partir d'une approche pratique, pour trouver une solution suffisamment bonne, c'est ce que l'on appelle une heuristique. Pour vulgariser, l'analyse de chaîne se fait en deux grandes étapes :

- Le repérage de caractéristiques connues ;
- La déduction d'hypothèses.

Un des objectifs de l'analyse de chaîne consiste à regrouper diverses activités sur Bitcoin en vue de déterminer l'unicité de l'utilisateur les ayant effectuées. Par la suite, il sera possible de tenter de rattacher ce faisceau d'activités à une identité réelle grâce à un point d'entrée. Il est important de comprendre que l'analyse de chaîne n'est pas une science exacte. Elle repose sur des heuristiques dérivées d'observations antérieures ou d'interprétations logiques. Ces règles permettent d'obtenir des résultats assez fiables, mais jamais d'une précision absolue. En d'autres termes, l'analyse de chaîne implique toujours une dimension de vraisemblance dans les conclusions émises. Par exemple, on pourra estimer avec plus ou moins de certitude que deux adresses appartiennent à une même entité, mais une certitude totale sera toujours hors de portée. Tout l'objectif de l'analyse de chaîne réside précisément dans l'agrégation de diverses heuristiques en vue de minimiser le risque d'erreur. Il s'agit en quelque sorte d'une accumulation de preuves qui nous permet de nous approcher davantage de la réalité. Ces fameuses heuristiques peuvent être regroupées en différentes catégories : *Les* patterns* de transaction (ou modèles de transaction) ;

- Les heuristiques internes à la transaction ;
- Les heuristiques externes à la transaction.

Notons que les deux premières heuristiques sur Bitcoin ont été formulées par Satoshi Nakamoto lui-même. Il les expose dans la partie 10 du *White Paper* (livre blanc). Il est intéressant d'observer que ces deux heuristiques conservent toujours une prééminence dans l'analyse de chaîne aujourd'hui. Ce sont la CIOH (*Common Input Ownership Heuristic*) et la réutilisation d'adresse.

Termes associés : **CIOH** | **HEURISTIQUE D'ANALYSE**

ANCESTOR MINING

PROTOCOLE | FR : MINAGE DES ANCÊTRES

Autre nom parfois donné à CPFP (*Child-Pay-For-Parent*). Le minage des ancêtres est le principe selon lequel un mineur ne choisit pas une transaction uniquement sur la base de ses propres frais de transaction, mais prend aussi en compte les frais des transactions ascendantes.

Terme associé : **CPFP - CHILD PAY FOR PARENT**

ANCHOR

RGB

Dans le cadre du protocole RGB, un *Anchor* représente un ensemble de données côté client permettant de prouver l'inclusion d'un engagement unique dans une transaction. Dans le protocole RGB, un *Anchor* est constitué des éléments suivants : *L'identifiant de la transaction Bitcoin (TXID) de la* witness transaction* ; *La preuve du* Multi Protocol Commitment* (MPC), comprenant la profondeur, le cofacteur et la preuve de Merkle ; *L'*Extra Transaction Proof *(ETP), qui fournit les données complémentaires nécessaires à la validation d'un engagement de type* Tapret *(non requise pour* Opret*).

Un *Anchor* sert donc à établir un lien vérifiable entre une transaction Bitcoin précise et des données privées validées par le protocole RGB. Il garantit que ces données sont bel et bien incluses dans la blockchain, sans pour autant que leur contenu exact soit exposé publiquement.

Termes associés : **DBC** | **RGB**

ANCHOR OUTPUTS

LIGHTNING NETWORK | FR : SORTIES D'ANCRAGE

Proposition qui vise à améliorer la gestion des frais de transaction dans le cadre des canaux Lightning. À chaque changement d'état dans un canal Lightning, les parties prenantes créent et signent une nouvelle transaction d'engagement qui reflète la nouvelle répartition des fonds au sein du canal. Le problème de ce mécanisme réside dans la détermination des frais de transaction au moment de sa création. En effet, les frais de transaction sur le réseau Bitcoin sont sujets à de fortes fluctuations, tant à la hausse qu'à la baisse. Si les frais fixés pour la dernière transaction d'engagement sont insuffisants au moment de la fermeture unilatérale du canal, non seulement la transaction prendra un temps considérable à se confirmer, mais les mécanismes de verrouillage temporel (*timelocks*) pourraient également permettre un vol des fonds. Les *anchor outputs* permettent de réserver une petite partie des fonds dans une transaction d'engagement pour couvrir les frais futurs. En cas de congestion du réseau et d'augmentation des frais, les *anchor outputs* permettent de modifier les frais de transaction après la création de la transaction d'engagement, garantissant ainsi une fermeture suffisamment rapide du canal Lightning.

Termes associés : **LIGHTNING NETWORK** | **TRANSACTION D'ENGAGEMENT**

ANCHORS.DAT

RÉSEAU

Fichier utilisé dans le client Bitcoin Core pour stocker les adresses IP des nœuds sortants de type *block-relay-only* auxquels un client était connecté avant d'être éteint. `anchors.dat` est donc créé à chaque fois que le nœud est arrêté et supprimé lorsqu'il est relancé. Les nœuds dont les adresses IP sont contenues dans ce fichier sont utilisés pour aider à établir rapidement des connexions lors du redémarrage du nœud.

Termes associés : **BITCOIN CORE** **PEERS.DAT**

ANCRAGE BILATÉRAL

SIDECHAIN | EN : 2WP - TWO-WAY PEG

Mécanisme qui permet d'établir une connexion entre le système principal de Bitcoin et une sidechain (ou une drivechain), c'est-à-dire une chaîne latérale. L'ancrage bilatéral assure une corrélation de valeur fixe entre les bitcoins sur la blockchain principale et les actifs correspondants sur la sidechain, permettant ainsi de déplacer des bitcoins entre les deux chaînes. Pour ce faire, les bitcoins sont temporairement verrouillés sur la blockchain principale et un montant équivalent d'actifs est émis sur la sidechain. Cela permet de profiter des avantages spécifiques de la sidechain, comme des transactions plus rapides ou des fonctionnalités de confidentialité améliorées, tout en maintenant la valeur des bitcoins utilisés. Lorsque les utilisateurs souhaitent revenir à la blockchain Bitcoin, le processus s'inverse : les actifs sur la sidechain sont détruits et les bitcoins correspondants sont déverrouillés. Il existe de nombreux mécanismes d'ancrages bilatéraux différents qui peuvent reposer sur :

- Un tiers de confiance unique ;
- Une fédération d'entités ;
- Les mineurs de la chaîne principale (drivechain).

Termes associés : **DRIVECHAIN** **SIDECHAIN**

ANNEX

PROTOCOLE | FR : ANNEXE

Champ optionnel réservé dans la structure de témoin (*witness*) des entrées SegWit v1 (Taproot), défini par le BIP-0341. L'annex occupe la dernière position dans la pile du témoin et se distingue par un préfixe `0×50`. Bien qu'il n'ait actuellement aucune fonction assignée par les règles de consensus, cet espace a été prévu dès la conception de Taproot pour accueillir de futures extensions du protocole.

Une propriété importante de l'annex est que toute signature Taproot ou Tapscript doit s'engager (*commit*) sur sa valeur lorsqu'il est présent. Cela signifie qu'un attaquant ne peut pas ajouter, modifier ou supprimer un annex sans invalider l'ensemble des signatures de l'entrée concernée. Cette garantie cryptographique assure que l'annex ne peut pas être exploité pour altérer une transaction déjà signée.

Termes associés : **TAPROOT** **TÉMOIN DE TRANSACTION**

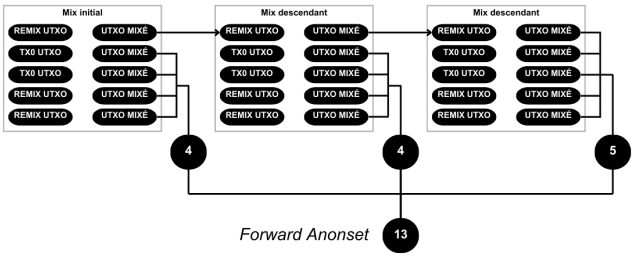
ANONSETS - ANONYMITY SETS

CONFIDENTIALITÉ	FR : ENSEMBLES D'ANONYMAT

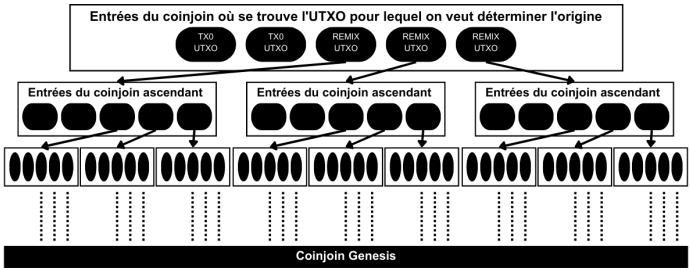
Les anonsets servent d'indicateurs pour évaluer le degré de confidentialité d'un UTXO particulier. Plus spécifiquement, ils mesurent le nombre d'UTXOs indistinguables au sein de l'ensemble qui inclut la pièce étudiée. Puisqu'il faut disposer d'un groupe d'UTXOs identiques, les anonsets sont généralement calculés au sein d'un cycle de coinjoins. Ils permettent, le cas échéant, de juger de la qualité des coinjoins. Un anonset de grande taille signifie un niveau d'anonymat accru, car il devient difficile de distinguer un UTXO spécifique au sein de l'ensemble. Deux types d'anonsets existent :

- L'ensemble d'anonymat prospectif ;
- L'ensemble d'anonymat rétrospectif.

Le premier indique la taille du groupe parmi lequel se cache l'UTXO étudié en sortie, sachant l'UTXO en entrée. Cet indicateur permet de mesurer la résistance de la confidentialité de la pièce face à une analyse passé vers présent (entrée vers sortie). En anglais, le nom de cet indicateur est « *forward anonset* », ou « *forward-looking metrics* ».



Le second indique le nombre de sources possibles pour une pièce donnée, sachant l'UTXO en sortie. Cet indicateur permet de mesurer la résistance de la confidentialité de la pièce face à une analyse présent vers passé (sortie vers entrée). En anglais, le nom de cet indicateur est « *backward anonset* », ou « *backward-looking metrics* ».



En français, il est globalement admis d'utiliser le terme « anonset ». On pourrait toutefois le traduire par « ensemble d'anonymat » ou « potentiel d'anonymat ».

En anglais et en français, on parle également parfois de « score » pour évoquer les anonsets (score prospectif et score rétrospectif).

Terme associé : **COINJOIN**

ANTI-EXFIL SIGNING

CRYPTOGRAPHIE | FR : SIGNATURE ANTI-EXFILTRATION

Protocole de signature qui empêche un hardware wallet compromis de faire fuiter la clé privée de l'utilisateur par le biais des signatures qu'il produit. Dans le schéma de signature ECDSA ou Schnorr, chaque signature nécessite un nonce aléatoire. Un appareil malveillant peut manipuler ce nonce pour y encoder discrètement des informations sur la clé privée, une attaque dite « par canal caché » (*covert channel*). L'attaque Dark Skippy a notamment montré qu'il est possible d'extraire l'intégralité d'une graine HD en seulement deux signatures.

Le protocole anti-exfiltration (aussi appelé « anti-kleptographie ») repose sur un échange de contributions au nonce entre l'hôte (le logiciel de portefeuille) et le hardware wallet. L'hôte fournit d'abord une valeur aléatoire au signataire matériel, qui doit l'intégrer dans le calcul de son nonce. Le nonce final résulte donc de la combinaison des deux contributions, ce qui empêche l'une ou l'autre partie de le contrôler seule. Grâce à un engagement cryptographique, l'hôte peut vérifier que sa contribution a bien été prise en compte dans la signature produite.

Ce mécanisme est implémenté dans des hardware wallets comme le BitBox02 et le Jade de Blockstream. Il constitue une défense essentielle contre les attaques de la chaîne d'approvisionnement, où le firmware d'un appareil pourrait avoir été modifié avant sa livraison.

Termes associés : **ECDSA** **MULTISIG**

ANTPOOL

MINAGE

Pool de minage de Bitcoin fondée en 2014 par Bitmain Technologies, le principal fabricant de matériel de minage ASIC. Basée en Chine, Antpool figure régulièrement parmi les plus grandes pools de minage au monde en termes de hashrate. Elle propose plusieurs méthodes de rémunération et s'adresse aussi bien aux mineurs individuels qu'aux infrastructures de minage professionnelles.

Termes associés : **MINAGE** **MINING POOL**

AOPP

PORTEFEUILLE

Sigle de « *Address Ownership Proof Protocol* ». C'est un protocole controversé, conçu pour prouver automatiquement la propriété d'adresses Bitcoin. Ce mécanisme permet aux utilisateurs de démontrer qu'ils contrôlent une adresse spécifique, directement à travers leur logiciel de portefeuille compatible. Initialement, l'AOPP a été créé pour simplifier la vérification de possession d'adresses, une exigence légale pour les clients désirant transférer leurs bitcoins hors des plateformes d'échange dans certaines juridictions, telles que la Suisse.

Néanmoins, ce protocole a été l'objet de critiques importantes au sein de la communauté Bitcoin, car il pourrait établir un précédent où les utilisateurs devraient demander l'autorisation pour exercer leur droit de possession sur leurs propres fonds (*self-custody*). Face à ces critiques, de nombreux logiciels de portefeuille ont choisi de ne pas adopter ce protocole.

Termes associés : **KYC - KNOW YOUR CUSTOMER** | **SELF-CUSTODY**

APERTURE

LIGHTNING NETWORK

Reverse proxy HTTP 402 développé par Lightning Labs, conçu pour permettre l'accès payant à des services web via le Lightning Network. Concrètement, Aperture se place devant un serveur web ou une API et exige un micropaiement Lightning avant d'autoriser l'accès à la ressource demandée.

Aperture repose sur le protocole L402 (*Lightning HTTP 402*), qui combine le code de statut HTTP 402 (« Payment Required »), les macaroons et le Lightning Network pour créer un standard d'authentification et de paiement sur le web. Lorsqu'un client envoie une requête vers une ressource protégée, Aperture répond avec un code 402 accompagné d'un macaroon et d'une facture Lightning. Une fois le paiement effectué, le client obtient une préimage qui, combinée au macaroon, forme un jeton L402 valide. Ce jeton sert ensuite à la fois d'authentification et de preuve de paiement pour les requêtes suivantes.

Aperture supporte le proxying de requêtes gRPC (HTTP/2) et REST (HTTP/1 et HTTP/2). Il intègre également un système de *rate limiting* par jeton ou par adresse IP, basé sur un algorithme de *token bucket*. Utilisé en production notamment par Lightning Loop, Aperture illustre une vision du web où les micropaiements remplacent les modèles traditionnels d'abonnement ou de publicité pour monétiser des API et des services en ligne.

Termes associés : **L402** | **MACAROON**

API

</> INFORMATIQUE | FR : INTERFACE DE PROGRAMMATION

Sigle de « *Application Programming Interface* ». Dans le contexte général de l'informatique, une API est un ensemble de règles et de spécifications que les logiciels peuvent suivre pour communiquer entre eux. Elles permettent aux développeurs d'accéder à des fonctionnalités ou à des données d'une application, d'un système d'exploitation ou d'un autre service pour leur propre logiciel.

Termes associés : **BITCOIN-CLI** | **RPC - REMOTE PROCEDURE CALL**

APO - ANYPREVOUT

BIP

Nom donné au BIP-0118 qui propose d'ajouter deux nouveaux SigHash Flag modificateurs, nommés `SIGHASH_ANYPREVOUT` et `SIGHASH_ANYPREVOUTANYSCRIPT`. Le terme « *AnyPrevOut* » provient de la contraction de « *Any Previous Output* » que l'on pourrait traduire en français par « toute sortie précédente ».

Termes associés : **BIP-0118** | **SIGHASH FLAG**

AQUA

PORTEFEUILLE

Portefeuille mobile développé par JAN3, disponible sur iOS et Android, conçu pour simplifier l'utilisation de Bitcoin, du Lightning Network et du réseau Liquid dans une seule application. Aqua permet d'envoyer et de recevoir des bitcoins on-chain ou sur la sidechain Liquid. Pour les paiements Lightning, Aqua recourt à des atomic swaps, ce qui lui permet de se passer d'un nœud Lightning embarqué tout en conservant la custody des fonds à tout moment.

Termes associés : **LIGHTNING NETWORK** | **LIQUID NETWORK**

ARBITRAGE

$ ÉCONOMIE ET RÉGULATION

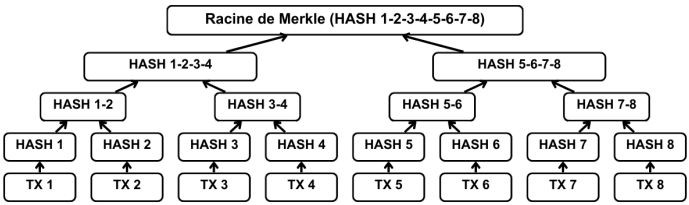
Pratique consistant à exploiter les différences de prix du BTC (ou de tout autre actif) entre différentes plateformes d'échange pour réaliser un profit. L'arbitrage implique d'acheter du bitcoin sur une plateforme où le prix est relativement bas et de le vendre simultanément sur une autre plateforme où le prix est plus élevé. Les écarts de prix peuvent survenir en raison de différences dans la liquidité, la demande, les volumes de transaction et les délais de transfert entre les plateformes d'échange. L'arbitrage contribue à équilibrer les prix sur différentes plateformes.

Termes associés : **BEAR MARKET** | **LIQUIDITÉS**

ARBRE DE MERKLE

CRYPTOGRAPHIE | EN : MERKLE TREE

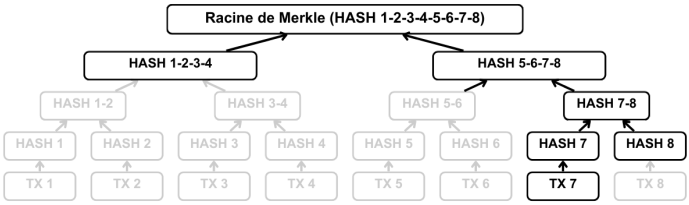
Un Arbre de Merkle est un accumulateur cryptographique. C'est une méthode pour justifier l'appartenance d'une information donnée à un ensemble plus grand. C'est une structure de données qui facilite la vérification d'informations dans un format compact. Dans le système Bitcoin, les arbres de Merkle sont utilisés pour regrouper et condenser les transactions d'un bloc en un unique hachage, appelé la racine de Merkle (ou « *Root Hash* »). Chaque transaction est hachée, puis les hachages adjacents sont hachés ensemble de façon hiérarchique jusqu'à ce que la racine de Merkle soit obtenue.



Cette structure permet de vérifier rapidement si une transaction spécifique est incluse dans un bloc donné sans avoir à analyser l'ensemble des transactions. Par exemple, si je dispose seulement de la racine de Merkle et que je souhaite vérifier que la `TX 7` fait bien partie de l'arbre, j'aurai uniquement besoin des preuves suivantes :

- `TX 7` ;
- `HASH 8` ;
- `HASH 5-6` ;
- `HASH 1-2-3-4`.

Grâce à ces quelques informations, je suis en capacité de calculer les nœuds intermédiaires jusqu'à la racine de Merkle.



Les arbres de Merkle sont notamment utilisés pour les nœuds légers (dits « SPV ») qui ne conservent que les entêtes de blocs, mais pas les transactions. On retrouve également cette structure dans le protocole UTREEXO, un protocole permettant de condenser l'UTXO set des nœuds, et dans le MAST Taproot.

L'arbre de Merkle porte le nom de Ralph Merkle, un cryptographe qui a conçu cette structure en 1979. Un arbre de Merkle peut également être nommé « arbre

de hachage ». En anglais, on dit « Merkle Tree » ou « Hash Tree ».

Termes associés : **MAST** | **TAPROOT**

ARK

COUCHE SUPÉRIEURE

Protocole de couche supérieure de type joinpool sur Bitcoin, proposé par Burak Keceli en mai 2023 et implémenté à partir de 2024 par Ark Labs. Ark permet de partager des UTXO on-chain entre plusieurs utilisateurs via des *Virtual Transaction Outputs* (VTXO) afin d'améliorer le passage à l'échelle de Bitcoin.

Le protocole s'articule autour d'*Ark Service Providers* (ASP), des opérateurs qui organisent périodiquement des « rounds » au cours desquels les VTXO des utilisateurs sont consolidés et ancrés dans de nouvelles transactions Bitcoin on-chain. Ce mécanisme empêche toute réserve fractionnaire. Les transferts entre utilisateurs d'un même ASP sont rapides, peu coûteux et offrent une confidentialité comparable à celle d'un coinjoin. Des paiements « hors round » sont également possibles pour un règlement immédiat, au prix d'une confiance temporaire envers l'ASP et l'expéditeur.

Le protocole est non-custodial, mais les VTXO expirent après une période définie. Si l'utilisateur ne se manifeste pas avant l'expiration, l'ASP peut récupérer la liquidité. L'utilisateur conserve la possibilité d'effectuer une sortie unilatérale on-chain à tout moment.

La version implémentée sans covenants, appelée *clArk*, émule les conditions de dépense par des multisignatures n-parmi-n entre participants d'un round, au prix d'une surcharge en bande passante et en stockage. L'ajout de covenants à Bitcoin améliorerait la scalabilité et réduirait l'interactivité requise. Ark est complémentaire du Lightning Network et interopérable via des passerelles.

Termes associés : **COINJOIN** | **LIGHTNING NETWORK** | **VTXO**

ASCII

INFORMATIQUE

Sigle de « *American Standard Code for Information Interchange* ». C'est un système de codage de caractères pour les ordinateurs. Le standard ASCII utilise 7 bits pour représenter 128 caractères différents : des lettres majuscules et minuscules de l'alphabet latin, des chiffres, des symboles de ponctuation, et des commandes de contrôle, comme le saut de ligne ou la tabulation.

Termes associés : **BIT** | **OCTET**

ASHIGARU

⊸ PORTEFEUILLE

Portefeuille Bitcoin mobile orienté vers la confidentialité, issu d'un fork du code de Samourai Wallet après la saisie des serveurs par les autorités américaines en avril 2024. Ashigaru reprend les fonctionnalités de protection de la vie privée de Samourai, notamment les coinjoins Whirlpool, les transactions Stonewall, les transactions collaboratives (Stowaway, StonewallX2), les ricochets et le contrôle avancé des UTXO. Le projet est maintenu par une communauté open source qui poursuit le développement de ces outils de confidentialité.

Termes associés : **COINJOIN** | **WHIRLPOOL**

ASIC

↗ MINAGE | FR : CIRCUIT INTÉGRÉ SPÉCIFIQUE À UNE APPLICATION

Un ASIC est un composant électronique conçu pour exécuter une fonction spécifique avec une efficacité optimale. Dans le contexte du minage de Bitcoin, les ASIC sont des circuits intégrés spécialisés qui effectuent des opérations de hachage à haute vitesse et faible consommation d'énergie. Ils sont spécialisés dans l'exécution de la fonction de hachage `SHA256` utilisée dans le mécanisme de la preuve de travail. L'ASIC est initialement le nom de la puce. Par extension, l'acronyme « ASIC » vise souvent à désigner également la machine qui héberge cette puce. Ainsi, les ordinateurs spécialisés dans le minage de Bitcoin sont parfois appelés des « ASIC », ou bien des « mineurs ». Les ASIC ont progressivement remplacé les autres méthodes de minage, telles que l'utilisation de processeurs (CPU) et de cartes graphiques (GPU), en raison de leur efficacité énergétique supérieure et de leur taux de hachage bien plus élevé.

Termes associés : **MINAGE** | **SHA256**

ASIC RESISTANCE

↗ MINAGE | FR : RÉSISTANCE AUX ASIC

Propriété d'un algorithme de minage qui vise à limiter l'avantage des circuits intégrés spécialisés (ASIC) par rapport au matériel grand public, comme les processeurs (CPU) ou les cartes graphiques (GPU). L'objectif est de maintenir une décentralisation du minage en empêchant que seuls des acteurs disposant de matériel coûteux et spécialisé puissent participer efficacement à la validation des blocs.

Pour atteindre cette résistance, certaines cryptomonnaies alternatives utilisent des fonctions de hachage qui requièrent beaucoup de mémoire vive (*memory-hard*), comme Scrypt ou Ethash, ce qui rend la fabrication d'ASIC plus difficile ou moins avantageuse. L'idée est que si l'algorithme exige des ressources variées (mémoire, bande passante, calcul), un ASIC n'offre pas un gain de performance aussi net que pour un algorithme simple comme SHA-256.

Dans le cas de Bitcoin, la résistance aux ASIC n'a jamais été un objectif. Bitcoin utilise SHA-256, une fonction qui se prête très bien à l'optimisation matérielle. Le réseau Bitcoin a donc vu apparaître des ASIC dès 2013, ce qui a progressivement rendu le minage par CPU et GPU non rentable. La résistance aux ASIC

reste un sujet de débat : ses partisans y voient un gage de décentralisation, tandis que ses détracteurs soulignent qu'elle ne fait que retarder la spécialisation du matériel sans jamais l'empêcher durablement.

Termes associés : **ASIC** | **CPU - CENTRAL PROCESSING UNIT**

ASICBOOST

↗ MINAGE

Méthode d'optimisation algorithmique inventée en 2016, conçue pour augmenter l'efficacité du minage de Bitcoin d'environ 20 % en réduisant la quantité de calculs nécessaires pour chaque tentative de hachage de l'entête. Cette technique exploite une particularité de la fonction de hachage SHA256, utilisée pour le minage, qui divise les données en blocs avant de les traiter. AsicBoost conserve l'un de ces blocs inchangé à travers plusieurs tentatives de hachage, ce qui permet au mineur de ne réaliser qu'une partie du travail pour ce bloc sur plusieurs tentatives. Ce partage de données permet une réutilisation des résultats de calculs précédents, ce qui diminue ainsi le nombre total de calculs nécessaires pour trouver un hachage valide.

AsicBoost peut être utilisé sous deux formes : Overt ASICBoost et Covert ASICBoost. La forme Overt AsicBoost est visible par tous, car elle implique d'utiliser le champ `nVersion` du bloc comme un nonce supplémentaire. La forme Covert cherche à masquer ces modifications en utilisant les arbres de Merkle. En revanche, cette seconde méthode n'est plus efficace depuis SegWit à cause du second arbre de Merkle qui multiplie le nombre de calculs nécessaires pour l'utiliser.

Pour résumer, AsicBoost permet de ne pas avoir à effectuer un vrai SHA256 complet pour toutes les tentatives de hachage, car une partie du résultat reste inchangée, ce qui permet d'accélérer le travail des mineurs.

Termes associés : **ASIC** | **SHA256**

ASMAP

🌐 RÉSEAU

Outil inventé par Gleb Naumenko et utilisé par Bitcoin Core pour améliorer la sécurité et la topologie du réseau Bitcoin en diversifiant les connexions entre les nœuds. Il s'agit d'une carte d'adressage IP vers les numéros de systèmes autonomes (ASN), permettant une meilleure répartition des connexions sortantes en fonction de l'ASN plutôt que des préfixes IP. Cela aide à prévenir les attaques Eclipse (notamment l'attaque Erebus) en rendant plus difficile pour un attaquant de monopoliser les connexions d'un nœud cible via des adresses IP appartenant au même système autonome.

Termes associés : **ECLIPSE** | **EREBUS**

ASP - ARK SERVICE PROVIDER

COUCHE SUPÉRIEURE

Opérateur du protocole Ark, responsable de la gestion des VTXO et de l'ancrage périodique des transactions sur la blockchain Bitcoin. L'ASP organise des « rounds » (toutes les 15 à 60 minutes selon la configuration) au cours desquels les transactions VTXO des utilisateurs sont consolidées et inscrites *on-chain*, empêchant toute réserve fractionnaire.

L'ASP est *non-custodial* : il ne détient pas directement les fonds des utilisateurs et ne peut donc pas les voler. Cependant, si un utilisateur ne se connecte pas pendant la période d'expiration de ses VTXO (typiquement 4 semaines), l'ASP peut récupérer la liquidité associée. Ce modèle oriente Ark vers un usage de paiements courants plutôt que de stockage à long terme. L'ASP assure également l'interopérabilité avec le Lightning Network via des passerelles, permettant aux utilisateurs Ark d'envoyer et recevoir des paiements Lightning.

Les ASP sont par nature plus volumineux que les *Lightning Service Providers* (LSP), ce qui pourrait favoriser une certaine centralisation du système. Pour atténuer ce risque, les ASP peuvent être fédérés ou fonctionner en pair-à-pair. Pour les institutions ou gros détenteurs de bitcoin, opérer un ASP constitue une opportunité de rendement avec un minimum de confiance requise, en fournissant la liquidité nécessaire au protocole.

Termes associés : **ARK** | **LIGHTNING SERVICE PROVIDER**

ASSET ID

COUCHE SUPÉRIEURE | FR : IDENTIFIANT D'ACTIF

Identifiant unique de 32 octets attribué à chaque actif émis via le protocole Taproot Assets. L'*asset ID* est produit en hachant avec SHA-256 cinq éléments concaténés : l'outpoint de la transaction de genèse (celle qui crée l'actif), un tag choisi par l'émetteur (par exemple le hash d'un nom de marque), le hash des métadonnées associées à l'actif (liens, images, documents), l'index de la sortie contenant le *commitment* Taproot Assets dans la transaction de genèse, et le type de l'actif (normal ou *collectible*).

```
asset_id=sha256(genesis_outpoint‖asset_tag‖asset_meta_hash‖outpu
t_index‖asset_type)
```

Puisqu'il dépend de l'outpoint de genèse, l'*asset ID* est unique à l'échelle mondiale : deux émissions distinctes produiront toujours des identifiants différents. Il sert de clé pour retrouver un actif dans un *Universe*, vérifier sa provenance et distinguer les différents types d'actifs engagés dans un même *Sparse Merkle Sum Tree*.

Termes associés : **ASSET MERGE** | **TAPROOT ASSETS PROTOCOL**

ASSET MERGE

COUCHE SUPÉRIEURE | FR : FUSION D'ACTIFS

Opération consistant à combiner deux actifs Taproot Assets du même type en un seul au sein d'une même sortie Taproot. Lors d'un *asset merge*, des actifs provenant de sorties Taproot distinctes sont regroupés dans un unique *Merkle-Sum Sparse Merkle Tree* engagé dans une nouvelle transaction. La somme totale à la racine de l'arbre reflète alors l'ensemble des actifs fusionnés. Cette opération est l'inverse d'un *asset split*. Elle permet de consolider des soldes fragmentés après plusieurs réceptions et de simplifier la gestion de l'arbre d'actifs d'un détenteur.

Termes associés : **ASSET SPLIT** | **TAPROOT ASSETS PROTOCOL**

ASSET PROOF

COUCHE SUPÉRIEURE | FR : PREUVE D'ACTIF

Preuve cryptographique permettant de vérifier l'existence, la propriété et la provenance d'un actif émis via le protocole Taproot Assets. Une *asset proof* contient le script d'actif ainsi que le chemin dans le *Sparse Merkle Sum Tree* menant à la feuille du détenteur concerné. Le destinataire peut reconstruire partiellement l'arbre, tweaker la clé publique de l'émetteur et vérifier que la transaction de genèse existe bien sur la blockchain Bitcoin.

Terme associé : **TAPROOT ASSETS PROTOCOL**

ASSET SPLIT

COUCHE SUPÉRIEURE | FR : DIVISION D'ACTIFS

Opération consistant à diviser un actif Taproot Assets en deux parties réparties dans des sorties Taproot distinctes. L'expéditeur met à jour le *Merkle-Sum Sparse Merkle Tree* de sa propre sortie en réduisant son solde, puis le destinataire construit un second arbre engagé dans une nouvelle sortie Taproot contenant les actifs reçus. Chaque transfert d'actifs entre détenteurs repose sur ce mécanisme de division. Les preuves d'actifs (*asset proofs*) doivent remonter jusqu'à la transaction de genèse pour attester la provenance des actifs après chaque *split*. C'est l'opération inverse d'un *asset merge*.

Termes associés : **ASSET MERGE** | **TAPROOT ASSETS PROTOCOL**

ASSIGNMENT

RGB

Dans la logique du protocole RGB, un Assignment est l'équivalent d'une sortie de transaction (output) qui modifie, met à jour ou crée certaines propriétés au sein de l'état d'un contract. Un Assignment comporte deux éléments :

- Une Seal Definition (la référence à un UTXO précis) ;
- Un Owned State (les données décrivant l'état associé à ce nouveau détenteur).

Un Assignment indique donc qu'une portion de l'état (par exemple, un actif) est désormais allouée à un détenteur particulier, identifié via un Single-use Seal lié

à un UTXO.

Termes associés : **RGB** | **SINGLE-USE SEAL**

ASSUME UTXO

PROTOCOLE

Fonctionnalité du client majoritaire Bitcoin Core qui permet à un nœud qui vient d'être initialisé (mais qui n'a pas encore fait l'IBD) de reporter la vérification des transactions et de l'UTXO set avant un snapshot donné. Le concept repose sur l'utilisation d'un UTXO set (liste de tous les UTXOs existants à un moment donné) chargé par l'utilisateur via la commande RPC `loadtxoutset` et présumé exact après vérification de son empreinte cryptographique par rapport à un hash hardcodé dans Core, ce qui permet au nœud d'être synchronisé très rapidement sur la chaîne avec le plus de travail accumulé. Puisque le nœud saute la longue étape de l'IBD, il est très rapidement fonctionnel pour son utilisateur. Assume UTXO divise la synchronisation (IBD) en deux parties : *Tout d'abord, le nœud réalise le* Header First Sync* (vérification des en-têtes seulement) et il considère comme valide l'UTXO set dont l'empreinte correspond au hash hardcodé dans Core ;

- Puis, une fois qu'il est fonctionnel, le nœud va vérifier l'historique complet des blocs en arrière-plan, en actualisant un nouvel UTXO set qu'il aura vérifié lui-même. Si ce dernier ne correspond pas à l'UTXO set initialement chargé, il fournira un message d'erreur.

Assume UTXO permet donc d'accélérer la préparation d'un nouveau nœud Bitcoin en reportant le processus de vérification des transactions et de l'UTXO set grâce à un snapshot dont le hash est hardcodé dans Core.

Termes associés : **ASSUME VALID** | **IBD - INITIAL BLOCK DOWNLOAD**

ASSUME VALID

PROTOCOLE

Paramètre de configuration dans le client majoritaire Bitcoin Core qui permet à un nœud qui vient d'être initialisé (mais qui n'a pas encore fait l'IBD) de sauter la vérification des signatures pour toutes les transactions incluses dans les blocs antérieurs à un certain bloc donné. Ce fameux bloc est défini par l'empreinte de son en-tête, c'est-à-dire son hash. Le bloc choisi est renouvelé lors de chaque nouvelle version de Bitcoin Core. À son initialisation, puisque ce paramètre est activé par défaut, le nœud va donc vérifier la chaîne d'en-têtes de blocs pour trouver la branche avec le plus de travail accumulé. Si le nœud détecte le hash fourni par Core dans la branche qu'il a retenue, il omettra la vérification des signatures pour les blocs antérieurs. Dans le cas contraire, le nœud procédera à une synchronisation traditionnelle (IBD) pour tout vérifier par lui-même.

L'objectif d'Assume Valid est d'accélérer le processus de synchronisation initiale d'un nœud sans compromettre la sécurité, en supposant que la majorité du réseau ait déjà validé ces transactions dans le passé. Le seul vrai compromis pour le nœud est qu'en cas de vol antérieur de bitcoins, il ne sera pas averti. Cependant, il peut toujours s'assurer de l'exactitude de la quantité de

bitcoins émis. Les nœuds poursuivent la vérification des signatures de transactions postérieures au bloc Assume Valid. Cette approche repose sur l'hypothèse que si une transaction est acceptée par le réseau depuis assez longtemps sans contestation, il est improbable qu'elle soit frauduleuse.

Termes associés : **ASSUME UTXO** | **IBD - INITIAL BLOCK DOWNLOAD**

ASYNC PAYMENTS

LIGHTNING NETWORK | FR : PAIEMENTS ASYNCHRONES

Mécanisme qui permet d'envoyer un paiement Lightning à un destinataire qui n'est pas en ligne au moment de l'envoi. Sur le réseau Bitcoin classique, les transactions sont naturellement asynchrones puisque le destinataire n'a pas besoin d'être connecté pour recevoir des fonds. En revanche, le protocole Lightning exige normalement que le receveur soit en ligne pour révéler le secret (la préimage) qui finalise le paiement via un HTLC.

Cette contrainte pose un problème important pour les portefeuilles mobiles, qui ne sont pas connectés en permanence. Plusieurs approches ont été proposées pour résoudre cette limitation. La plus prometteuse repose sur l'utilisation des PTLC (*Point Time-Locked Contracts*) combinés avec des factures statiques BOLT-12, qui permettent au payeur de construire un paiement valide sans interaction préalable avec le destinataire. Une autre approche utilise des nœuds relais trampoline : un nœud intermédiaire de confiance retient le paiement jusqu'à ce que le destinataire se reconnecte, puis lui transmet.

Des implémentations concrètes ont vu le jour, notamment dans LDK et Eclair. LDK a intégré la logique de paiements asynchrones côté client et côté serveur, tandis qu'Eclair permet de réveiller un pair mobile déconnecté pour lui transmettre un paiement en attente. Ces avancées sont essentielles pour rendre Lightning véritablement utilisable sur les terminaux mobiles, où la connectivité permanente ne peut pas être garantie.

Termes associés : **GOSSIP** | **HTLC**

ATH - ALL-TIME HIGH

ÉCONOMIE ET RÉGULATION | FR : PLUS HAUT HISTORIQUE

Désigne le niveau le plus élevé jamais atteint par l'élément étudié. Souvent, l'ATH désigne le plus haut niveau de prix du bitcoin en comparaison avec une monnaie étatique sur une période donnée.

Termes associés : **BULL RUN** | **HALVING**

ATLC

COUCHE SUPÉRIEURE

Sigle de « *Anchor Timelock Contracts* ». C'est un paiement conditionnel utilisé dans le cadre du protocole Ark pour fournir un calendrier de paiement atomique à un hub, grâce à des connecteurs permettant de former ce que l'on appelle un « txlock ». L'objectif d'un ATLC est sensiblement le même que celui d'un HTLC sur Lightning.

Terme associé : **ARK**

ATOMIC MULTI-PATH PAYMENTS

LIGHTNING NETWORK | FR : PAIEMENTS MULTICHEMINS ATOMIQUES

Version améliorée des MPP (*Multi-Path Payments*) où chaque fragment de paiement possède un secret partiel distinct, ce qui garantit que la transaction est réglée de manière atomique, c'est-à-dire en entier ou pas du tout.

Les MPP sont des techniques de paiement sur Lightning qui permettent de fractionner une transaction en plusieurs petites parties pour les acheminer via différentes routes. Autrement dit, chaque fraction de paiement emprunte un chemin de nœuds différent. Cela permet de contourner les limitations de liquidité sur un canal unique dans la route. Dans les MPP de base, chaque fraction de paiement partage le même secret, et donc le même hash dans les HTLCs. Cela peut rendre la transaction traçable si un observateur est présent sur plusieurs routes, car il peut faire un lien entre tous ces secrets identiques. De plus, du fait que le secret est unique pour toutes les fractions du paiement, celui-ci n'est pas atomique. Cela signifie que certaines parties du paiement peuvent être exécutées avec succès, tandis que d'autres peuvent échouer.

Dans les AMP, on utilise des secrets partiels uniques pour chaque fraction. Une fois tous les fragments reçus, ils permettent au destinataire de reconstituer le secret général d'origine et de réclamer l'intégralité du paiement. Cette méthode rend le règlement partiel du paiement impossible, car la possession de tous les secrets partiels est nécessaire pour débloquer le paiement complet. Cela garantit que le paiement est entièrement réussi ou entièrement échoué (c'est-à-dire, atomique). Les AMP améliorent aussi la confidentialité, car il n'y a plus de liens visibles entre les différentes routes.

Un avantage des AMP est qu'ils fonctionnent même si seuls le receveur et l'envoyeur ont implémenté cette méthode. Les nœuds intermédiaires traitent ces paiements comme des transactions classiques en utilisant des HTLCs, sans être conscients qu'ils traitent des fractions d'un paiement plus important.

On parle également parfois de « Atomic Multi-Part Payment » pour désigner cette même méthode.

Terme associé : **MPP - MULTI-PATH PAYMENTS**

ATOMIC SWAP

OUTIL | FR : ÉCHANGE ATOMIQUE

Technologie permettant un échange de cryptomonnaies directement entre deux parties, sans besoin de confiance et sans nécessiter d'intermédiaire. Ces échanges sont dits « atomiques » car ils ne peuvent donner que deux résultats :

- Soit l'échange réussit et les deux participants se sont effectivement échangé leurs cryptomonnaies ;
- Soit l'échange échoue et les deux participants repartent avec leurs cryptomonnaies de départ.

Les Atomic Swaps peuvent s'effectuer soit avec une même cryptomonnaie, dans ce cas, on parle également de « *coinswap* », soit entre des cryptomonnaies différentes. Historiquement, ils s'appuyaient sur des « *Hash Time-Locked*

Contracts » (HTLC), un système de verrouillage temporel qui garantit la complétude ou l'annulation totale de l'échange, préservant ainsi l'intégrité des fonds des parties impliquées. Cette méthode exigeait des protocoles capables de gérer à la fois les scripts et les timelocks. Toutefois, ces dernières années, la tendance s'est orientée vers l'utilisation des *Adaptor Signatures*. Cette seconde approche présente l'avantage de se passer de scripts, ce qui réduit ainsi les coûts opérationnels. Son autre atout majeur réside dans le fait qu'elle n'exige pas l'emploi d'un hachage identique pour les deux volets de la transaction, ce qui permet d'éviter de révéler un lien entre elles.

Termes associés : **ADAPTOR SIGNATURE** | **HTLC**

ATTAQUE DE FINNEY

ATTAQUE | EN : FINNEY ATTACK

Type de double dépense qui cible les transactions à zéro confirmation. L'attaquant, qui doit être mineur, pré-mine un bloc qui contient une transaction de son adresse A vers une autre adresse B qu'il contrôle également. Au lieu de diffuser ce bloc, il effectue un paiement depuis cette même adresse A vers l'adresse C d'un commerçant. Le commerçant voit la transaction dans le mempool, ne détecte aucun conflit et livre la marchandise. L'attaquant diffuse alors son bloc pré-miné : la transaction vers B est confirmée, tandis que le paiement vers C est invalidé.

Hal Finney a décrit cette attaque en 2011 sur le forum BitcoinTalk. Elle se distingue de l'attaque des 51 % car elle ne nécessite pas une puissance de calcul majoritaire : il suffit que l'attaquant trouve un bloc, même avec un hashrate modeste. En contrepartie, sa fenêtre d'exécution est étroite et elle ne fonctionne que contre les commerçants qui acceptent des paiements sans attendre de confirmation.

Termes associés : **DOUBLE DÉPENSE** | **FINNEY HAL**

ATTAQUE DES 51%

ATTAQUE | EN : 51% ATTACK

Scénario hypothétique sur le système Bitcoin où un acteur malveillant contrôle plus de 50 % de la puissance de calcul totale du minage (*hashrate*). Avec une telle dominance, l'attaquant peut manipuler le processus de consensus, permettant des actions malveillantes telles que la double dépense, où les mêmes bitcoins sont dépensés une première fois sur une chaîne finalement rendue désuète, puis une seconde fois sur la chaîne valide. Une autre finalité d'une attaque des 51 % est la censure des transactions. Cependant, réaliser une attaque des 51 % nécessite des ressources financières, humaines, énergétiques et techniques considérables, et rend l'acteur malveillant susceptible d'être découvert avant que l'attaque n'ait lieu. Bien que théoriquement possible, une attaque des 51 % sur Bitcoin est considérée comme très peu probable en raison de la décentralisation du minage et de la grande puissance de calcul actuellement déployée.

Cette attaque est également nommée « Attaque Goldfinger ».

Termes associés : **DOUBLE DÉPENSE** | **HASHRATE**

AUTOLOOP

LIGHTNING NETWORK

Fonctionnalité de Lightning Loop qui automatise les opérations de *Loop In* et de *Loop Out* en fonction de seuils prédéfinis. L'opérateur d'un nœud Lightning configure des limites de liquidité entrante et sortante pour ses canaux, et Autoloop déclenche automatiquement des *submarine swaps* lorsque ces seuils sont dépassés. Cela permet de maintenir un équilibre de liquidité sans intervention manuelle, ce qui est particulièrement utile pour les nœuds de routage qui doivent constamment ajuster la répartition de leurs fonds entre canaux pour maximiser les revenus de routage.

Termes associés : **LOOP** **LOOP IN**

AVERAGE ROUND DURATION

MINAGE | FR : DURÉE MOYENNE DE TOUR

La durée moyenne de tour est un indicateur utilisé pour estimer le temps nécessaire à une pool de minage pour trouver un bloc, en fonction de la difficulté du réseau et du hashrate de la pool. Il est calculé en prenant le nombre de shares attendues pour trouver un bloc et en le divisant par le hashrate de la pool. Par exemple, si une pool de minage compte 200 mineurs, et que chacun génère en moyenne 4 shares par seconde, la puissance totale de calcul de la pool est de 800 shares par seconde :

```
200 * 4 = 800
```

En supposant qu'il faille, en moyenne, produire 1 million de shares pour trouver un bloc valide, l'*Avg. Round Duration* de la pool sera de 1 250 secondes, soit environ 21 minutes :

```
1 000 000 / 800 = 1 250
```

Concrètement, cela signifie qu'en moyenne, la pool de minage devrait trouver un bloc toutes les 21 minutes environ. Cet indicateur fluctue avec les variations du hashrate de la pool : une augmentation du hashrate réduit l'*Avg. Round Duration*, tandis qu'une diminution l'allonge. Il va également fluctuer à chaque évolution périodique de la cible de difficulté sur Bitcoin (tous les 2016 blocs). Cette mesure ne prend pas en compte les blocs trouvés par d'autres pools et se concentre uniquement sur la performance interne de la pool étudiée.

Termes associés : **LUCK** **SHARES**

B

B-MONEY

HISTOIRE

Concept de cryptomonnaie décentralisée proposé par Wei Dai en 1998. Ce système imaginait un réseau où les participants seraient identifiés uniquement par des clés publiques, et où chaque transaction serait signée par l'expéditeur. B-money était établi sur un modèle de comptes, contrairement à Bitcoin qui utilise un modèle d'UTXO. Il permettait la création de monnaie par une sorte de preuve de travail liée à un panier de marchandises. C'était donc un précurseur du principe actuel de stablecoin. Ce concept n'a jamais été mis en œuvre.

Termes associés : **BIT GOLD** | **ECASH - DAVID CHAUM**

BANLIST.DAT

RÉSEAU

Nom de l'ancien fichier utilisé par le logiciel Bitcoin Core pour enregistrer les adresses IP des nœuds qui ont été bannis par l'utilisateur. Depuis la version 22.0, on utilise le fichier `banlist.json` à la place.

Termes associés : **BANLIST.JSON** | **PEERS.DAT**

BANLIST.JSON

RÉSEAU

Nom du fichier utilisé par le logiciel Bitcoin Core pour enregistrer les adresses IP des nœuds qui ont été bannis par l'utilisateur. Ce fichier contient donc une liste des nœuds bannis avec lesquels le nœud ne se connectera pas. Cette fonctionnalité permet d'empêcher les interactions avec des nœuds potentiellement nuisibles ou malveillants.

Termes associés : **BANLIST.DAT** | **BITCOIN CORE**

BARE-MULTISIG

PROTOCOLE | FR : MULTISIG BRUT

Modèle de script standard P2MS utilisé pour établir des conditions de dépense sur un UTXO. Il permet de bloquer des bitcoins à l'aide de plusieurs clés publiques. Pour dépenser ces bitcoins, il faut fournir une signature avec un nombre prédéfini de clés privées associées. Par exemple, un P2MS `2/3` dispose de `3` clés publiques avec `3` clés privées secrètes associées. Pour dépenser les bitcoins bloqués avec ce script P2MS, il faut réaliser une signature avec au moins `2` parmi les `3` clés privées. C'est un système de sécurisation à seuil (*threshold*). Ce script a été inventé en 2011 par Gavin Andresen alors qu'il venait de récupérer la maintenance du client principal de Bitcoin. Aujourd'hui, le P2MS n'est utilisé qu'à la marge par certaines applications. L'extrême majorité des multisignatures modernes emploient d'autres modèles de scripts comme le P2SH ou le P2WSH. Par rapport à ceux-ci, le P2MS est extrêmement trivial. Les clés publiques le constituant sont dévoilées dès la réception de la transaction. L'utilisation d'un P2MS est également plus chère que les autres scripts multisignatures.

Termes associés : **P2MS** | **P2SH**

BASE - ARITHMÉTIQUE

</> INFORMATIQUE

Une base est un système de numération qui utilise un nombre fixe de caractères pour représenter tous les nombres possibles. La base détermine le nombre de symboles distincts disponibles pour représenter les chiffres dans ce système. Par exemple, le système le plus connu dans nos vies quotidiennes est la base 10, également appelée système décimal. Elle utilise dix symboles distincts `(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)` pour représenter tous les nombres. D'autres systèmes de numération couramment utilisés dans les domaines de l'informatique et des mathématiques incluent le système binaire (base 2), avec deux symboles `(0, 1)`, et le système hexadécimal (base 16), avec seize symboles `(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)`. Dans le cadre de Bitcoin, vous rencontrerez parfois des encodages en base 58 ou en base 32 adaptée (nommée `Bech32`).

Termes associés : **BINAIRE** | **HEXADÉCIMAL**

BASE FEE

LIGHTNING NETWORK | FR : FRAIS DE BASE

Composante fixe des frais de routage sur le Lightning Network. Lorsqu'un nœud transmet un paiement pour le compte d'un autre, il peut prélever des frais composés de deux parties : le *base fee*, un montant constant facturé pour chaque paiement routé indépendamment de son montant, et le *fee rate*, une part proportionnelle au montant transféré. Le *base fee* est exprimé en millisatoshis et vaut typiquement 1 000 msat (soit 1 satoshi). Il est configurable par l'opérateur du nœud pour chaque canal.

Termes associés : **FEE RATE - LIGHTNING** | **MILLISATOSHI**

BASE58CHECK

PROTOCOLE

La `Base58Check` est un encodage utilisé dans le système Bitcoin pour représenter les adresses de réception Legacy et certaines autres données, telles que les clés étendues, sous forme de chaînes de caractères lisibles par l'homme. C'est une variante du système `Base58`, une représentation positionnelle de base 58 conçue pour minimiser les erreurs de transcription humaine. Elle utilise un ensemble de 58 caractères alphanumériques, composé des chiffres de `1` à `9`, des lettres majuscules de `A` à `Z` (à l'exception des lettres `I` et `O` pour éviter la confusion avec les chiffres `1` et `0`) et des lettres minuscules de `a` à `z` (à l'exception de la lettre `l` pour éviter la confusion avec le chiffre `1`). La `Base58Check` se distingue de la `Base58` par l'ajout d'une somme de contrôle (*checksum*). Elle est représentée par une version réduite d'un double hachage `SHA256` des données originales (`SHA256d` ou `HASH256`), à la fin des données encodées en `Base58`. Lors de la vérification, la somme de contrôle est recalculée et comparée à celle qui a été ajoutée lors de l'encodage. Si les deux empreintes correspondent, les données sont considérées comme valides, sinon une erreur de corruption ou de transcription est signalée.

L'utilisation de la `Base58Check` dans les adresses Bitcoin et les clés procure plu-

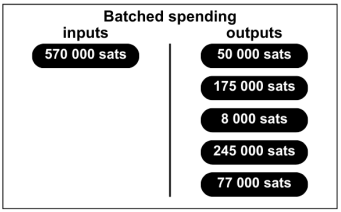
sieurs avantages. Premièrement, elle permet de réduire les erreurs humaines lors de la transcription et de la lecture en évitant les caractères ambigus. Deuxièmement, elle protège contre les erreurs de saisie en détectant et signalant les erreurs grâce à la *checksum*. Troisièmement, la représentation compacte des données en `Base58Check` permet de réduire l'espace requis pour stocker et partager les adresses et les clés. Les adresses de réception les plus récentes (post-SegWit) ont abandonné cet encodage `Base58Check` pour des encodages `Bech32` et `Bech32m`, disposant d'une somme de contrôle plus évoluée (avec des codes BCH).

Termes associés : **ADRESSE DE RÉCEPTION** | **BECH32 ET BECH32M**

BATCHED SPENDING

PROTOCOLE | FR : DÉPENSES GROUPÉES

Technique de dépense employée principalement par les entités ayant un volume élevé de transactions, comme les plateformes d'échange par exemple, pour optimiser et réduire les coûts de transaction. En regroupant plusieurs paiements destinés à différents destinataires en une seule transaction Bitcoin, la dépense groupée permet de consommer moins d'espace dans les blocs, ce qui permet de réduire les frais associés. Le batched spending se distingue par un modèle facilement reconnaissable lors d'une analyse de chaîne. Ce modèle se manifeste par l'utilisation de quelques UTXO en entrée (souvent un seul) et la création de multiples UTXO en sortie. L'interprétation de ce modèle est que nous sommes en présence d'une dépense groupée. Face à ce modèle en analyse de chaîne, on peut en déduire que l'UTXO en entrée provient d'une société avec une grosse activité économique et que les UTXO en sorties vont se disperser. Certains appartiendront à des clients de la société. D'autres iront peut-être vers des sociétés partenaires. Enfin, il y aura certainement un change qui reviendra à la société émettrice.



Termes associés : **ANALYSE DE CHAINE** | **UTXO**

BBQR

</> INFORMATIQUE

Sigle de « Better Bitcoin QR ». Protocole qui permet d'encoder des fichiers trop volumineux pour un seul QR code en une série de QR codes successifs (QR animé). Les types de données cibles sont les PSBT (BIP-0174) et les transactions signées, mais le format prend aussi en charge les données CBOR, JSON et texte.

Le protocole est adopté par plusieurs logiciels et *signing devices*, notamment COLDCARD Q, Sparrow Wallet et Nunchuk. La spécification est publiée par Coinkite en tant que standard ouvert.

Termes associés : **PSBT** **SEEDQR**

BCH - BITCOIN CASH

HISTOIRE

Système de cryptomonnaie issu d'un *hard fork* de Bitcoin (BTC) réalisé le 1er août 2017 au bloc 478 558. Ce fork est survenu à la suite de désaccords au sein de la communauté Bitcoin concernant les solutions à adopter pour résoudre les problèmes de passage à l'échelle du protocole (Blocksize War). Alors que Bitcoin a implémenté SegWit (*soft fork*) qui comprend une légère augmentation détournée de la capacité des blocs, Bitcoin Cash a opté pour une augmentation directe de la taille des blocs (*hard fork*), passant de 1 Mo à 8 Mo, avec l'objectif de réduire les frais de transaction et d'améliorer les temps de confirmation.

Termes associés : **BLOCKSIZE WAR** **SEGWIT**

BCH CODE

CRYPTOGRAPHIE | FR : CODE BCH

Classe de codes de correction d'erreurs utilisés pour détecter et corriger des erreurs dans une séquence de données. Autrement dit, les codes de correction d'erreur BCH permettent de trouver et de corriger des erreurs aléatoires dans des informations transmises, afin de garantir qu'elles arrivent intactes à leur destination. Le sigle « BCH » représente les initiales des noms des inventeurs de ces codes : Bose, Ray-Chaudhuri, et Hocquenghem.

On utilise cet outil dans de nombreux domaines de l'informatique, notamment dans les SSD, les DVD ou encore les QR codes. Par exemple, grâce à ces codes de correction d'erreurs, même si un QR code est partiellement recouvert, il est toujours possible de récupérer les informations qu'il encode.

Dans le cadre de Bitcoin, les codes BCH sont utilisés pour la somme de contrôle dans les formats d'adresses Bech32 et Bech32m (SegWit natif). Ils représentent un meilleur compromis entre la taille et la capacité de détection d'erreurs de la checksum par rapport aux simples fonctions de hachage utilisées précédemment sur les adresses Legacy. Dans le contexte de Bitcoin, les codes BCH sont utilisés uniquement pour la détection d'erreurs, et non pour

la correction d'erreur. Ainsi, les logiciels de portefeuille Bitcoin identifieront et signaleront à l'utilisateur toute erreur dans une adresse de réception, mais ne modifieront pas l'adresse incorrecte automatiquement. Ce choix est motivé par le fait que l'intégration de la correction d'erreur affecte inévitablement les capacités de détection d'erreur de l'algorithme.

Termes associés : **BASE58CHECK** | **BECH32 ET BECH32M**

BCOIN

PROTOCOLE

Implémentation alternative du protocole Bitcoin développée en JavaScript pour Node.js. Contrairement à Bitcoin Core, qui est l'implémentation majoritaire de nœuds Bitcoin (environ 99 % du réseau), Bcoin est plutôt orienté vers les besoins des entreprises et des développeurs qui cherchent à intégrer certaines fonctionnalités liées à Bitcoin dans leurs systèmes. Bcoin inclut un ensemble de bibliothèques complètes pour intégrer un nœud complet, un nœud SPV, un portefeuille HD, un indexeur, et d'autres fonctionnalités.

Termes associés : **BITCOIN CORE** | **IMPLÉMENTATION DE BITCOIN**

BDK - BITCOIN DEV KIT

OUTIL

Kit de développement (SDK) pour les portefeuilles sur Bitcoin. BDK est une collection de bibliothèques et d'outils destinés aux développeurs, permettant de simplifier la création d'applications de portefeuilles Bitcoin. BDK fournit des modules de fonctionnalités essentielles telles que la gestion de portefeuilles, la construction de transactions, la signature de transactions ou encore la gestion des clés. Les développeurs peuvent ensuite s'appuyer sur ces modules pour concevoir leurs propres logiciels. Les composants de BDK sont élaborés dans un souci de légèreté et de modularité, afin de les rendre ajustables à la plupart des possibilités d'utilisation. L'objectif de cet outil est de centraliser le développement de portefeuilles Bitcoin afin de concentrer les efforts.

BDK était auparavant appelé « Magical Bitcoin ».

Termes associés : **LDK - LIGHTNING DEV KIT** | **PORTEFEUILLE**

BEAR MARKET

ÉCONOMIE ET RÉGULATION | FR : MARCHÉ BAISSIER

Période prolongée durant laquelle le prix d'un actif, tel que le bitcoin, diminue par rapport à une monnaie fiat, typiquement le dollar américain. Depuis son introduction en 2009, le cycle des bear markets du BTC semble influencé par les halvings, des événements programmés qui réduisent de moitié la récompense des mineurs et qui interviennent approximativement tous les quatre ans.

Termes associés : **BULL MARKET** | **HALVING**

BECH32 ET BECH32M

PROTOCOLE

`Bech32` et `Bech32m` sont deux formats d'encodage d'adresse pour recevoir des bitcoins. Ils sont établis sur une base 32 légèrement modifiée. Ils embarquent une somme de contrôle établie sur un algorithme de correction d'erreurs appelé BCH (*Bose-Chaudhuri-Hocquenghem*). Par rapport aux adresses Legacy, encodées en `Base58check`, les adresses `Bech32` et `Bech32m` disposent d'une somme de contrôle plus performante, permettant de détecter et potentiellement de localiser les fautes de frappe. Leur format dispose également d'une meilleure lisibilité, avec uniquement des caractères minuscules. Voici la matrice d'addition de ce format depuis la base 10 :

+	0	1	2	3	4	5	6	7
0	q	p	z	r	y	9	x	8
8	g	f	2	t	v	d	w	0
16	s	3	j	n	5	4	k	h
24	c	e	6	m	u	a	7	l

`Bech32` et `Bech32m` sont des formats d'encodage utilisés pour représenter les adresses SegWit. `Bech32` est un format d'encodage d'adresse introduit par la BIP-0173 en 2017. Il utilise un ensemble de caractères spécifiques, composé de chiffres et de lettres minuscules, pour minimiser les erreurs de frappe et faciliter la lecture. Les adresses `Bech32` commencent généralement par `bc1` pour indiquer qu'elles sont natives de SegWit. Ce format est uniquement utilisé sur les adresses SegWit V0, avec les scripts P2WPKH (*Pay to Witness Public Key Hash*) et P2WSH (*Pay to Witness Script Hash*). Toutefois, il existe une petite faille inattendue propre au format `Bech32`. Chaque fois que le dernier caractère de l'adresse est un `p`, l'ajout ou la suppression d'un nombre quelconque de caractères `q` le précédant immédiatement n'invalide pas la somme de contrôle. Cela n'affecte pas les utilisations existantes des adresses SegWit V0 (BIP-0173) en raison de leur restriction à deux longueurs définies. Cependant, cela pourrait affecter des utilisations futures de l'encodage `Bech32`. Le format `Bech32m` est simplement un format `Bech32` avec cette erreur rectifiée. Il a été introduit avec le BIP-0350 en 2020. Les adresses `Bech32m` commencent également par `bc1`, mais elles sont spécifiquement conçues pour être compatibles avec la version SegWit V1 (Taproot) et les versions ultérieures, avec le script P2TR (*Pay to TapRoot*).

Termes associés : **BASE58CHECK** | **BCH CODE**

BEHIND-THE-METER

MINAGE | FR : DERRIÈRE LE COMPTEUR

Dans le cadre du minage de Bitcoin, désigne l'électricité produite et directement distribuée à une ferme, sans passer par le réseau public.

Termes associés : **ASIC** | **MINAGE**

BERKELEYDB

PROTOCOLE

Système de gestion de base de données intégrable avec une architecture de stockage clé-valeur. Il a été utilisé dans les premières versions de Bitcoin et a été remplacé par LevelDB en 2013.

Termes associés : **BITCOIN CORE** | **LEVELDB**

BETTERHASH

MINAGE

Protocole de minage développé par Matt Corallo en 2018 pour essayer de contrer la centralisation croissante du minage sur les pools. Il se distingue de Stratum, qui est alors la norme, en offrant aux hacheurs plus de contrôle sur la sélection des transactions à inclure dans les *block templates*. L'idée principale derrière BetterHash est de redonner aux hacheurs la possibilité de gérer eux-mêmes la construction du *block template*, tout en conservant les avantages des pools de minage comme la réduction de la variance des revenus.

Dans le protocole Stratum, les pools de minage contrôlent la création des *block templates*, c'est-à-dire qu'ils choisissent quelles transactions sont incluses dans les blocs et déterminent également la chaîne sur laquelle miner. Cette centralisation affaiblit Bitcoin, car cela rend le processus de confirmation des transactions vulnérable à la censure.

BetterHash permet aux hacheurs de reprendre la main sur ces opérations, tout en laissant la pool gérer la distribution des récompenses. C'est donc en quelque sorte un des précurseurs de Stratum V2.

Termes associés : **STRATUM** | **STRATUM V2**

BFT - BYZANTINE FAULT TOLERANCE

ATTAQUE | FR : TOLÉRANCE AUX DÉFAUTS BYZANTINS

Concept utilisé dans les systèmes distribués pour assurer un consensus fiable et sécurisé, même en présence de nœuds défectueux ou malveillants.

Ce concept est lié au problème des généraux byzantins, une métaphore utilisée pour illustrer les défis de la coordination et de la prise de décision dans un système distribué où les acteurs peuvent ne pas se faire confiance.

L'algorithme de preuve de travail (Proof-of-Work ou PoW) utilisé par Bitcoin est un mécanisme de consensus dit « Tolérant aux pannes byzantines » ou « Byzantine Fault Tolerant (BFT) ».

Termes associés : **PREUVE DE TRAVAIL** | **PROBLÈME DES GÉNÉRAUX BYZANTINS**

BGP HIJACKING

ATTAQUE | FR : DÉTOURNEMENT BGP

Attaque où un acteur malveillant manipule les annonces BGP pour rediriger le trafic Internet vers son propre réseau. En prétendant être l'origine légitime de certaines plages d'adresses IP, l'attaquant peut intercepter, surveiller ou bloquer le trafic destiné à ces adresses.

BGP (*Border Gateway Protocol*) est le protocole qui détermine comment le trafic est routé entre les différents réseaux autonomes (AS) qui composent l'Internet mondial. Il décide des chemins que les données prennent pour aller d'un point à un autre.

Dans le contexte de Bitcoin, le BGP Hijacking peut être utilisé pour détourner le trafic entre les nœuds du réseau Bitcoin pour en isoler certains. Cette attaque peut également être réalisée sur des serveurs Stratum utilisés par les pools de minage, afin de rediriger la puissance de calcul des hacheurs vers sa propre pool. Étant donné que les hacheurs ne redémarrent pas souvent leurs machines, cette redirection peut persister longtemps. Cette attaque peut non seulement viser un gain financier direct en usurpant les récompenses de bloc, mais également chercher à prendre temporairement plus de pouvoir sur le système.

Termes associés : **ATTAQUE DES 51%** | **ECLIPSE**

BIG BLOCKERS

HISTOIRE

Nom donné aux partisans de l'augmentation de la taille des blocs durant la Blocksize War entre 2015 et 2017. Les big blockers pensent qu'il est nécessaire d'augmenter cette limite pour permettre le passage à l'échelle de Bitcoin. Ils soutiennent des propositions comme Bitcoin XT, Bitcoin Classic et Bitcoin Unlimited. Ils n'ont pas de problème avec le fait d'implémenter des changements via des hard forks, contrairement aux small blockers qui préfèrent les soft forks.

Terme associé : **BLOCKSIZE WAR**

BIG-ENDIAN

</> INFORMATIQUE | FR : GROS-BOUTISTE

Format de stockage de données dans les systèmes informatiques où les octets les plus significatifs (les « gros bouts ») sont placés en premier dans l'ordre des adresses. Cela signifie que dans une séquence avec plusieurs octets, l'octet ayant le plus grand poids (par exemple, les chiffres les plus à gauche en hexadécimale) est stocké en premier.

Termes associés : **LITTLE-ENDIAN** | **OCTET**

BINAIRE

</> INFORMATIQUE | EN : BINARY

Système de numération en base 2, qui utilise le `0` et le `1` pour représenter l'information. Chaque chiffre dans ce système est appelé un bit, qui est l'abréviation de « binary digit ». Dans le contexte informatique, le binaire est la forme de représentation des données en mémoire. Les ordinateurs utilisent le binaire pour traiter toutes sortes d'informations, car il correspond aux deux états possibles d'un circuit électronique : éteint ou allumé.

Termes associés : **BASE - ARITHMÉTIQUE** | **BIT**

BINOHASH

SCRIPT

Fonction de hachage résistante aux collisions pour les transactions Bitcoin, proposée par Robin Linus (ZeroSync) en 2026, qui permet une forme limitée d'introspection de transaction dans Bitcoin Script sans modification du consensus. Le nom vient du coefficient binomial $\binom{n}{t}$ qui permet sa construction.

Le problème résolu par Binohash est que Bitcoin Script ne peut pas lire nativement les données d'une transaction (entrées, sorties, montants). Cette limitation empêche la vérification *trustless* des propriétés d'une transaction dans Script, ce qui est nécessaire pour des protocoles comme les ponts BitVM.

Binohash exploite le comportement *FindAndDelete* du legacy `OP_CHECKMULTISIG`, un artéfact historique qui supprime les signatures du *scriptCode* avant le calcul du *sighash*. En intégrant un pool de signatures dans le *locking script* et en sélectionnant des sous-ensembles via `OP_ROLL`, chaque sélection produit un *sighash* différent. Le dépenseur effectue un *grinding* par preuve de travail (puzzles de signatures ECDSA vérifiables par `OP_SIZE`) jusqu'à trouver un sous-ensemble satisfaisant une contrainte de taille. Les indices du sous-ensemble gagnant constituent le Binohash, directement lisible dans l'*unlocking script*.

Termes associés : **BITVM** | **COVENANT**

BIP

BIP | FR : PROPOSITION D'AMÉLIORATION DE BITCOIN

Acronyme de « *Bitcoin Improvement Proposal* ». Un BIP est un document formel proposant une amélioration, une spécification technique ou une information relative au protocole Bitcoin et à son écosystème. Étant donné que Bitcoin ne possède pas d'entité centrale pour décider des mises à jour, les BIPs permettent à la communauté de suggérer, discuter et mettre en œuvre des améliorations de manière structurée et transparente.

Chaque BIP détaille les objectifs de l'amélioration proposée, les justifications, les impacts potentiels sur la compatibilité, ainsi que les avantages et inconvénients. Les BIPs peuvent être rédigés par n'importe quel membre de la communauté. Ils sont principalement la propriété de leurs auteurs et représentent leur recommandation personnelle : un BIP individuel ne constitue pas un consensus communautaire ni une recommandation générale d'implémentation.

Les éditeurs qui maintiennent le dépôt GitHub des BIPs sont : Bryan Bishop, Jon Atack, Luke Dashjr, Mark Erhardt (Murch), Olaoluwa Osuntokun et Ruben Somsen. Leur rôle est administratif et éditorial : ils vérifient le formatage, la pertinence et la qualité des propositions, mais n'évaluent pas si une proposition sera adoptée. Si quelqu'un propose une amélioration qui n'est pas acceptée dans le cadre formel des BIPs, il peut toujours la présenter directement à la communauté Bitcoin, voire créer un fork incluant sa modification. L'avantage du processus des BIPs réside dans sa formalité et sa centralisation, qui facilitent le débat pour éviter la division parmi les utilisateurs de Bitcoin, en cherchant à implémenter des mises à jour de manière consensuelle. À la fin, c'est bien le principe de majorité économique qui détermine les jeux de pouvoir au sein du protocole.

Les BIPs sont classés en trois types (depuis le BIP-0003 qui a remplacé le BIP-0002 en 2025) :

- `Specification BIPs` (anciennement « Standards Track ») : Définissent un ensemble de règles techniques décrivant une nouvelle fonctionnalité ou affectant l'interopérabilité des implémentations. Ils peuvent être implémentés et les implémentations peuvent être conformes à leur spécification. Ils doivent contenir une référence d'implémentation et des vecteurs de test avant d'atteindre le statut « Complete » ;
- `Informational BIPs` : Décrivent un problème de conception, fournissent des recommandations générales ou des informations utiles à la communauté Bitcoin ;
- `Process BIPs` : Décrivent un processus entourant Bitcoin ou proposent un changement de processus. Ils sont traités comme des documents vivants, modifiables indéfiniment par consensus approximatif sur la liste de diffusion.

Les BIPs de type « Specification » et « Informational » peuvent être classifiés par couche (« Layer ») :

- `Consensus (soft fork)` et `Consensus (hard fork)` : Concernent les règles de consensus de Bitcoin, telles que les modifications des règles de validation des blocs ou des transactions. Par exemple, les BIPs de SegWit et Taproot appartiennent à cette catégorie ;
- `Peer Services` : Concerne le fonctionnement du réseau de nœuds Bitcoin, c'est-à-dire comment les nœuds se trouvent et communiquent entre eux sur Internet ;
- `API/RPC` : Concerne les interfaces de programmation applicative (API) et les appels de procédure à distance (RPC) qui permettent aux logiciels Bitcoin d'interagir avec les nœuds ;
- `Applications` : Concerne les applications construites par-dessus Bitcoin. Les BIPs de cette catégorie vont typiquement traiter les modifications au niveau des standards des portefeuilles Bitcoin.

Le processus de soumission d'un BIP commence par la conceptualisation et la discussion de l'idée sur la liste de diffusion *Bitcoin Development Mailing List*. Les auteurs doivent d'abord vérifier que l'idée n'a pas déjà été proposée, puis la présenter publiquement pour recueillir un premier retour. Si l'idée est promet-

teuse et a suscité un intérêt suffisant dans la communauté, l'auteur rédige un BIP en respectant un format spécifique (MediaWiki ou Markdown) et le soumet via une Pull Request sur le dépôt GitHub des BIPs. Les éditeurs vérifient que la proposition respecte le formatage, est pertinente, a fait l'objet d'une discussion préalable sur la liste de diffusion, et est techniquement réalisable. Un numéro n'est attribué par un éditeur qu'après que la proposition a suscité un intérêt tangible dans la communauté. Les auteurs peuvent également nommer des « Deputies », des co-propriétaires du BIP qui les assistent dans le processus.

Les BIPs passent par quatre statuts au cours de leur cycle de vie (simplifiés par le BIP-0003 depuis les neuf statuts du BIP-0002) :

- Brouillon (`Draft`) : La proposition est en cours de rédaction et de discussion. Un BIP en « Draft » sans activité pendant un an peut être fermé si ses auteurs ne manifestent pas leur intention de poursuivre ;
- Complet (`Complete`) : Les auteurs ont terminé le travail prévu, la proposition est claire, complète et prête pour adoption. Les BIP de type « Specification » doivent inclure une implémentation de référence et des vecteurs de test. Après ce statut, les modifications doivent rester mineures. C'est le statut final de la plupart des BIPs informationnels réussis ;
- Déployé (`Deployed`) : Le BIP est activement utilisé dans l'écosystème. Ce statut est atteint sur présentation de preuves d'utilisation effective (déploiement en production par un projet établi, critères d'activation d'un soft fork remplis, etc.). Une fois déployé, le BIP est considéré comme final et ne devrait plus subir de modifications substantielles ;
- Fermé (`Closed`) : Le BIP n'est plus activement développé ni utilisé et ne présente qu'un intérêt historique. Ce statut remplace les anciens statuts « Deferred », « Rejected », « Withdrawn », « Replaced » et « Obsolete ».

Après avoir atteint le statut « Complete », tout changement doit être documenté dans une section « Changelog » et versionné selon un schéma sémantique.

Termes associés : **BIP-0001** **SOFT FORK**

BIP-0001

BIP

Document qui définit le processus d'élaboration et de mise en œuvre des améliorations proposées au protocole Bitcoin. Rédigé par Amir Taaki en septembre 2011, il établit une méthodologie standard pour proposer et documenter des modifications potentielles. Les propositions de BIP sont alors classées en trois catégories :

- `Standards Track` : ceux qui concernent les modifications directes du protocole Bitcoin et son interopérabilité ;
- `Informational` : ceux qui offrent des informations générales, mais n'affectent pas directement le protocole ;
- `Process` : ceux qui introduisent des changements non techniques, comme les procédures et lignes directrices.

Ce cadre systématise le développement de Bitcoin, afin d'assurer une approche

coordonnée et transparente de ses évolutions. La méthodologie du BIP-0001 sera par la suite remplacée par celle du BIP-0002.

Termes associés : **BIP** **BIP-0002**

BIP-0002

BIP

Document rédigé par Luke Dashjr en 2016 qui révise le processus des BIP (*Bitcoin Improvement Proposals*) défini par le BIP-0001. Le BIP-0002 clarifie les statuts possibles des propositions (« *Draft* », « *Proposed* », « *Final* », « *Deferred* », « *Withdrawn* », « *Rejected* », « *Replaced* », « *Obsolete* », « *Active* ») et précise les critères objectifs pour progresser d'un statut à l'autre. Il introduit notamment des règles concrètes pour qu'un BIP atteigne le statut « *Final* » : par exemple, un soft fork nécessite une majorité claire de mineurs, tandis qu'un BIP de couche applicative doit être implémenté par au moins deux logiciels indépendants. Le BIP-0002 ajoute également un système de commentaires publics, une classification par couche (*layer*) issue du BIP-0123, et des exigences de licence pour les nouvelles propositions. Ce processus a depuis été remplacé par celui du BIP-0003. Son statut est « *Closed* ».

Termes associés : **BIP-0001** **MÉTHODE D'ACTIVATION**

BIP-0003

BIP

Révision majeure du processus de BIP réalisée en 2024 et adoptée en 2026, remplaçant le BIP-0002 en vigueur depuis 2016. Le BIP-0003 simplifie le workflow en réduisant les neuf statuts précédents à quatre : « *Draft* », « *Complete* », « *Deployed* » et « *Closed* ». Il redéfinit les types de BIP en remplaçant « *Standards Track* » par « *Specification* », clarifiant ainsi que tous les BIP sont des recommandations individuelles de leurs auteurs et non des standards imposés. Le rôle des éditeurs BIP est recentré sur des tâches administratives et éditoriales, avec moins de jugements discrétionnaires sur l'acceptation des propositions.

Parmi les nouveautés, le BIP-0003 introduit un système de versionnage sémantique avec un « *Changelog* » obligatoire après le statut « *Complete* », la possibilité de nommer des « *Deputies* » pour assister les auteurs, et l'utilisation d'expressions SPDX pour les licences. Le format *Markdown* est désormais accepté en plus du *MediaWiki*. Le système de commentaires du BIP-0002, jugé inefficace par manque de participation, est aboli. Les BIP de type « *Process* » deviennent des documents vivants modifiables indéfiniment par consensus approximatif. Le BIP-0003 a le statut « *Deployed* ».

Termes associés : **BIP** **BIP-0001**

BIP-0008

BIP

Élaboré suite aux débats sur SegWit, qui utilisait le BIP-0009 pour son activation, le BIP-0008 est une méthode d'activation de soft forks qui incorpore nativement un mécanisme d'UASF (*User-Activated Soft Fork*) automatique. Comme le BIP-0009, le BIP-0008 utilise la signalisation des mineurs, mais ajoute le paramètre `LOT` (*Lock-in On Time out*). Si `LOT` est réglé sur `vrai`, à l'expiration de la période de signalisation sans atteindre le seuil requis, un UASF est automatiquement déclenché, forçant l'activation du soft fork. Cette approche contraint les mineurs à être coopératifs ou risquer un UASF imposé par les utilisateurs. De plus, contrairement au BIP-0009, le BIP-0008 définit la période de signalisation établie sur la hauteur des blocs, éliminant les manipulations potentielles via le taux de hachage par les mineurs. Le BIP-0008 permet également de fixer un seuil de vote variable et introduit un paramètre pour une hauteur de bloc minimale pour l'activation, donnant aux mineurs le temps de se préparer et de signaler leur accord en avance sans forcément être prêts. Lorsqu'un soft fork est activé via le BIP-0008 avec le paramètre `LOT=vrai`, on utilise ici une méthode très agressive contre les mineurs qui sont immédiatement mis sous la pression d'un éventuel UASF. En effet, ça leur laisse seulement 2 choix :

- Être coopératifs, et ainsi faciliter le processus d'activation ;
- Être non coopératifs, auquel cas les utilisateurs font un UASF automatiquement pour imposer le soft fork.

Termes associés : **BIP-0009** **UASF**

BIP-0009

BIP

Méthode d'activation de soft forks sur Bitcoin proposée en 2015. Elle introduit un système où les mineurs signalent leur soutien à un soft fork en utilisant un bit spécifique dans le champ de version des blocs. Un soft fork proposé sous BIP-0009 est activé si 95 % des blocs sur une période de 2016 blocs (environ deux semaines, coïncidant avec chaque ajustement de difficulté) signalent leur approbation. Après ce verrouillage, un délai est accordé pour que les mineurs se préparent à la mise à jour avant son activation. En cas d'échec à atteindre le seuil de 95 % dans la durée maximale prévue, le soft fork est abandonné. Le BIP-0009 permet la signalisation de plusieurs soft forks simultanément, mais donne un pouvoir considérable aux mineurs, car si le seuil requis n'est pas atteint, le soft fork est simplement abandonné. Cette méthode était celle initialement utilisée pour SegWit, avant que le BIP-0148 qui suggère l'utilisation d'un UASF ne vienne rebattre les cartes et forcer le verrouillage via le BIP-0091.

Termes associés : **BIP-0008** **MASF**

BIP-0010

BIP

Le BIP-0010, proposé par Alan Reiner en 2011, visait à standardiser un format d'échange pour la distribution et la collecte de signatures dans les transactions multisignatures. Plutôt que d'introduire le multisig lui-même, ce BIP proposait un protocole de communication appelé « *Transaction Distribution Proposal* » (TxDP) encodé en ASCII, permettant aux différentes parties d'une transaction multisig de se transmettre la transaction proposée, de collecter les signatures nécessaires, puis de diffuser la transaction complétée. Ce format incluait la totalité des transactions précédentes référencées, permettant ainsi à un dispositif de signature hors-ligne de vérifier les montants sans accès à la blockchain. Le BIP-0010 a été implémenté dans le logiciel Armory, mais a ensuite été abandonné au profit de formats plus récents. Son statut est « *Withdrawn* ».

Termes associés : **BIP-0011** **BIP-0016**

BIP-0011

BIP

BIP introduit par Gavin Andresen en 2011 qui propose une méthode standard pour créer des transactions multisignatures sur Bitcoin. Cette proposition permet d'améliorer la sécurité des bitcoins en exigeant plusieurs signatures pour valider une transaction. Le BIP-0011 introduit un nouveau type de script, nommé « M-of-N multisig », où `M` représente le nombre minimum de signatures requises parmi `N` clés publiques potentielles. Ce standard exploite l'opcode `OP_CHECKMULTISIG`, déjà existant dans Bitcoin, mais qui n'était pas conforme aux règles de standardisation des nœuds auparavant. Bien que ce type de script ne soit plus couramment utilisé pour des portefeuilles multisig réels, au profit du P2SH ou du P2WSH, son utilisation reste possible. Il est notamment utilisé dans des méta-protocoles tels que les Stamps. Les nœuds peuvent toutefois choisir de ne pas relayer ces transactions P2MS avec le paramètre `permitbaremultisig=0`.

De nos jours, on connait ce standard sous le nom de « bare-multisig » ou « P2MS ».

Termes associés : **BARE-MULTISIG** **MULTISIG**

BIP-0012

BIP

Proposition de Gavin Andresen pour améliorer la flexibilité et la confidentialité des scripts de transaction Bitcoin. Ce BIP propose d'implémenter un nouvel opcode de script, nommé `OP_EVAL`, qui permet d'évaluer un script contenu dans les données d'un `scriptSig` lors du processus de validation de la transaction. La principale modification du BIP-0012 est de permettre l'inclusion d'un script à l'intérieur d'un autre script. Cette technique permet la création de conditions plus complexes pouvant être vérifiées lors de la dépense, sans avoir besoin de les révéler aux utilisateurs qui envoient des bitcoins vers l'adresse utilisée. Le BIP-0012 a été ultérieurement remplacé par d'autres propositions plus sûres, notamment le BIP-0016 (P2SH), qui offre une méthode différente pour atteindre les mêmes objectifs que `OP_EVAL`.

Termes associés : **BIP-0016** **OP_EVAL**

BIP-0013

BIP

Présente une méthode standardisée pour créer les adresses P2SH. Le BIP-0013 spécifie le format d'adresse P2SH, qui commence par le préfixe `3`, et qui inclut le hachage d'un script plutôt que le hachage d'une clé publique. Ce type d'adresse restera longtemps le standard préféré pour les portefeuilles multisig.

Termes associés : **MULTISIG** **P2SH**

BIP-0014

BIP

BIP proposé par Patrick Strateman et Amir Taaki en 2011 qui vise à distinguer les numéros de version des clients et du protocole. Le BIP-0014 précise la façon dont les implémentations du protocole Bitcoin doivent se présenter sur le réseau. Il suggère l'utilisation d'un format d'agent-utilisateur pour identifier la version et le type du client Bitcoin utilisé. L'objectif principal du BIP-0014 est de faciliter la gestion des modifications et la détection des incompatibilités entre les différents clients existants. Alors qu'il était auparavant logique de considérer le client de Satoshi comme de fait le protocole Bitcoin, la multiplication des logiciels à cette période amène le BIP-0014 à bien différencier les clients du protocole lui-même.

Termes associés : **BIP-0001** **BITCOIN CORE**

BIP-0015

BIP

Proposition d'un système d'alias permettant d'associer des noms lisibles par l'humain à des adresses Bitcoin, via différents mécanismes de résolution. Le BIP-0015 visait à simplifier l'expérience utilisateur en remplaçant les longues adresses alphanumériques par des identifiants mémorisables de type `alias@domaine.com`. Quatre mécanismes étaient proposés : FirstBits (utilisant la blockchain comme annuaire), les enregistrements DNS TXT, la résolution via une requête HTTPS, et Namecoin (via un système de nommage décentralisé). Ce BIP est un précurseur conceptuel du BIP-0353 qui reprend l'idée d'adresses lisibles via DNS. Il n'a jamais été implémenté et a été fermé.

Termes associés : **BIP** **BIP-0021**

BIP-0016

BIP

Le BIP-0016 a introduit le concept de *Pay-to-Script-Hash* P2SH (en français « payer au hachage du script »). Proposé initialement en janvier 2012 puis activé en avril 2012, le BIP-0016 visait à simplifier l'utilisation de transactions nécessitant des scripts complexes, tels que les transactions multisignatures, en permettant aux utilisateurs de payer vers un *hash* du script requis pour dépenser ces bitcoins plutôt que le script lui-même. Cette innovation a permis de réduire la quantité de données nécessaires dans la transaction initiale, déplaçant la charge de fournir le script complet vers la partie qui dépense les bitcoins. Il a également permis de ne révéler le script qu'à la dépense des bitcoins engagés sur le script, plutôt que dès la réception. Le BIP-0016 revêt une importance historique puisqu'il symbolise l'une des premières modifications majeures du protocole Bitcoin après le retrait de Nakamoto en 2011. Ce BIP a été le centre de débats très tendus qui ont même poussé Gavin Andresen, successeur de Satoshi en tant que mainteneur principal, à s'octroyer une période de retrait. De nombreuses autres propositions existaient et certaines ont même failli être activées à la place du BIP-0016.

Termes associés : **BIP-0017** **P2SH**

BIP-0017

BIP

Proposition de Luke Dashjr concurrente au BIP-0012 et au BIP-0016. Le BIP-0017 introduisait un nouvel opcode, `OP_CHECKHASHVERIFY`, conçu pour permettre la vérification d'un script présent dans le `scriptSig` face à son hachage présent dans le `scriptPubKey` avant de débloquer les fonds. Le BIP-0016 (P2SH) a finalement été préféré au BIP-0017 (CHV) suite à une période de débats intenses.

Termes associés : **BIP-0016** **OP_CHECKHASHVERIFY - CHV**

BIP-0018

BIP

Proposition de Luke Dashjr intitulée « *hashScriptCheck* » qui visait à remplacer les champs `scriptSig` et `scriptPubKey` des transactions par trois nouveaux éléments : `dataSig` (données pures poussées sur la pile), `scriptCheck` (le script de validation) et `hashScriptCheck` (le hachage du script attendu placé dans l'output). Contrairement au BIP-0017 du même auteur qui proposait un nouvel opcode `OP_CHECKHASHVERIFY`, le BIP-0018 adoptait une approche plus radicale en redéfinissant la structure même des entrées et sorties de transactions. Sur le plan protocolaire, le résultat était identique au BIP-0016 (P2SH), mais le BIP-0018 formalisait la dépréciation explicite de `scriptSig` et `scriptPubKey`.

Termes associés : **P2SH** **SCRIPT**

BIP-0019

BIP

Proposition d'une variante de transactions *multisig* M-of-N avec un coût réduit en opérations de vérification de signatures (*sigops*). Le BIP-0019 a été supplanté par P2SH (BIP-0016) qui offre une solution plus générale et plus flexible pour encapsuler des scripts complexes, y compris le *multisig*, dans une adresse de hachage de script.

Termes associés : **MULTISIG** **P2SH**

BIP-0020

BIP

Première version du schéma d'URI `bitcoin:` pour faciliter les paiements via des liens cliquables. Le BIP-0020 proposait notamment un système de montants avec exposants (par exemple `50X8` pour 50×10^8 satoshis). Il a été remplacé par le BIP-0021, rédigé par Nils Schneider et Matt Corallo sur la base du travail de Luke Dashjr, qui a supprimé ces montants alternatifs et étendu la spécification avec des paramètres supplémentaires.

Termes associés : **BIP** **BIP-0021**

BIP-0021

BIP

Proposition rédigée par Nils Schneider et Matt Corallo, sur la base du BIP-0020 rédigé par Luke Dashjr, qui venait lui-même d'un autre document rédigé par Nils Schneider. Le BIP-0021 définit comment les adresses de réception doivent être encodées dans les URI (*Uniform Resource Identifier*) pour faciliter les paiements. Par exemple, une URI Bitcoin suivant le BIP-0021 dans laquelle je demanderais sous le label « *Pandul* » que l'on m'envoie 0.1 BTC ressemblerait à cela :

```
bitcoin:bc1qmp7emyf7un49eaz0nrxk9mdfrtn67v5y866fvs?amount=0.1&la
bel=Pandul
```

Cette standardisation améliore l'expérience utilisateur des transactions Bitcoin

en permettant de cliquer sur un lien ou de scanner un QR code pour lancer les paramètres de celles-ci. La norme BIP-0021 est aujourd'hui largement adoptée dans les logiciels de portefeuilles Bitcoin.

Termes associés : **ADRESSE DE RÉCEPTION** | **URI**

BIP-0022

BIP

BIP proposé en 2012 par Luke Dashjr qui introduit une méthode standardisée JSON-RPC pour les interfaces de minage externes, appelée `getblocktemplate`. Avec l'augmentation de la difficulté de minage, l'utilisation de logiciels externes spécialisés dans la production de preuves de travail s'est développée. Ce BIP propose une norme commune de communication pour le *block template* entre les nœuds complets et les logiciels spécialisés dans le minage. Cette méthode implique d'envoyer la structure entière du bloc, plutôt que simplement l'entête, afin de laisser la possibilité au mineur de la personnaliser.

Termes associés : **BIP-0023** | **BLOCK TEMPLATE**

BIP-0023

BIP

Ce BIP est une extension du BIP-0022, visant à encourager son adoption par les *pools* de minage. Proposée par Luke Dashjr, cette extension étend la norme `getblocktemplate` du BIP-0022 pour couvrir l'ensemble des fonctionnalités du protocole *Getwork*, ancêtre de *Stratum*, afin de faciliter la transition des *pools* depuis *Getwork* vers `getblocktemplate`. L'objectif principal du BIP-0023 est de transmettre l'intégralité du *block template* aux mineurs, leur permettant ainsi d'auditer et éventuellement de modifier la structure du bloc proposé par la *pool*. Cette démarche vise à atténuer les préoccupations liées à la centralisation du minage, en donnant aux mineurs individuels un plus grand contrôle et une meilleure transparence sur le processus de création des blocs.

Termes associés : **BIP-0022** | **GETWORK**

BIP-0030

BIP

Proposition d'amélioration impliquant un soft fork mis en œuvre le 15 mars 2012 afin de résoudre le problème des identifiants de transaction dupliqués. Avant le BIP-0030, il était techniquement possible d'avoir deux transactions différentes avec le même identifiant de transaction (TXID) dans la blockchain. C'est notamment arrivé deux fois pour des transactions coinbase : celle dans le bloc 91 880 dispose du même TXID que la coinbase du bloc 91 722, et celle dans le bloc 91 842 dispose du même TXID que la coinbase du bloc 91 812. Le BIP-0030 a résolu cette faille en imposant une nouvelle règle simple : aucune nouvelle transaction ne peut avoir le même TXID qu'une transaction précédemment enregistrée, à moins que la transaction originale n'ait été complètement dépensée (c'est-à-dire que tous ses outputs ont été utilisés). Ce soft fork a été

activé grâce à la méthode du *flag day*. C'est donc un des premiers UASF.

Termes associés : **BIP-0034** **COINBASE**

BIP-0031

BIP

Proposition visant à améliorer les mécanismes de gestion du réseau par les nœuds Bitcoin. Avant le BIP-0031, les nœuds Bitcoin n'avaient pas de moyen direct de savoir si leurs pairs étaient toujours connectés, fonctionnels et non surchargés. Le BIP-0031 a introduit l'utilisation d'un message `pong`, en réponse au message `ping`, qui permet une vérification active de la connectivité entre les nœuds.

Terme associé : **NOEUD**

BIP-0032

BIP

Le BIP-0032 a introduit le concept de portefeuille hiérarchique et déterministe (HD wallet). Cette proposition permet de générer une hiérarchie de paires de clés à partir d'une graine commune, la *master seed*, en utilisant des fonctions de dérivation à sens unique. Chaque paire de clés générée peut elle-même être le parent d'autres clés enfants, formant ainsi une structure arborescente (hiérarchique). L'avantage du BIP-0032 est qu'il permet de gérer de multiples paires de clés différentes avec la nécessité de ne conserver qu'une seule graine pour les régénérer. Cette innovation a notamment permis de lutter contre le phénomène de réutilisation d'adresse qui est grave pour la confidentialité des utilisateurs. Le BIP-0032 permet aussi la création de sous-branches au sein d'un même portefeuille afin de faciliter sa gestion.

Termes associés : **DÉRIVATION** **GRAINE**

BIP-0033

BIP

Proposition d'un système de nœuds « stratifiés » (*stratized nodes*) où les clients légers délèguent la validation et les recherches dans la blockchain à des services spécialisés (*blockchain services*), plutôt que d'effectuer ces opérations eux-mêmes. Le BIP-0033 visait à réduire la charge de validation pour les clients sur appareils peu puissants ou portables, en leur permettant de s'appuyer sur plusieurs services externes et de n'accepter une information que si un sous-ensemble commun de ces services la confirme. Cette approche n'a jamais été implémentée sur Bitcoin.

Termes associés : **NOEUD ÉLAGUÉ** **RÉSEAU BITCOIN**

BIP-0034

BIP

Soft fork appliqué en mars 2013, qui a introduit la version 2 pour les blocs Bitcoin. Le dernier bloc en version 1 est le bloc 227 835. Cette nouvelle version exige que chaque bloc inclue dans le `scriptSig` de la transaction coinbase la hauteur du bloc en cours de création. Cette modification permet d'une part de clarifier la manière dont le réseau consent à modifier la structure des blocs et les règles de consensus. D'autre part, cela permet de forcer l'unicité de chaque bloc et de chaque transaction coinbase.

Termes associés : **BIP-0030** **COINBASE**

BIP-0035

BIP

Proposition permettant à un nœud Bitcoin d'ouvrir les informations relatives à sa mempool, c'est-à-dire les transactions en attente de confirmation. Grâce à cela, d'autres acteurs peuvent recevoir des données en temps réel sur les transactions non confirmées en adressant un message spécifique à un nœud. Avant l'adoption du BIP-0035, les nœuds ne pouvaient accéder qu'aux informations concernant les transactions déjà confirmées. Cette amélioration offre aux portefeuilles SPV la possibilité de recevoir des informations sur les transactions non confirmées, permet à un mineur de ne pas omettre des transactions avec des frais élevés lors d'un redémarrage, et facilite l'analyse des informations de la mempool par des services externes.

Termes associés : **MEMPOOL** **NOEUD SPV - NOEUD LÉGER**

BIP-0036

BIP

Proposition d'un mécanisme permettant aux nœuds Bitcoin d'annoncer et de fournir des services personnalisés via le protocole pair-à-pair. Le BIP-0036 visait à étendre les capacités du réseau en permettant à des nœuds spécialisés d'offrir des fonctionnalités supplémentaires au-delà du relais standard de blocs et de transactions. Cette proposition n'a jamais été adoptée.

Termes associés : **BIP** **RÉSEAU BITCOIN**

BIP-0037

BIP

Proposition introduite pour permettre aux portefeuilles légers (*Simplified Payment Verification*) de filtrer les transactions sans avoir à télécharger la blockchain complète. Cette méthode repose sur le concept de Bloom Filters, des structures de données probabilistes qui sont utilisées pour tester l'appartenance à un ensemble. Ces filtres permettent aux clients SPV de recevoir uniquement les transactions pertinentes pour leur portefeuille, afin de réduire la bande passante et la mémoire vive requise pour la synchronisation, notamment sur les téléphones portables. Les Bloom Filters sont transmis à un nœud complet, lequel renvoie en réponse des « Merkle blocks », contenant uniquement les transactions filtrées, l'en-tête avec la racine de Merkle, et les branches nécessaires pour associer ces transactions à la racine de l'arbre de Merkle. Le BIP-0037 a été critiqué pour ses lacunes en matière de confidentialité, car il expose notamment les adresses et les transactions des utilisateurs aux nœuds complets utilisés. Pour tenter de remédier à cette faille, il est possible d'intégrer des transactions supplémentaires, les « faux positifs », afin de créer du déni plausible. Néanmoins, le volume de faux positifs nécessaire pour atteindre un niveau de déni plausible satisfaisant reste considérablement élevé. Par ailleurs, le BIP-0037 a aussi été critiqué pour la charge de calcul imposée aux nœuds complets et pour l'introduction d'un nouveau vecteur d'attaque de type DoS. Cette option est donc par défaut désactivée dans Bitcoin Core. Pour l'activer, il faut entrer le paramètre `peerbloomfilters=1` dans le fichier de configuration.

Terme associé : **BLOOM FILTER**

BIP-0038

BIP

Proposition d'amélioration de Bitcoin qui introduit un mécanisme de chiffrement pour ajouter une protection supplémentaire sur les clés privées grâce à une passphrase. Le BIP-0038 permet de garantir que même si un tiers obtient physiquement la clé privée chiffrée, il ne pourra pas l'utiliser sans connaître sa passphrase. Cela ajoute une couche de sécurité supplémentaire pour protéger des bitcoins contre le vol, notamment pour la sécurité des vieux *paper wallets*.

Bien que l'on désigne cette proposition sous le terme de « passphrase », il est important de ne pas la confondre avec la passphrase présentée dans le BIP-0039, cette dernière étant d'ailleurs bien plus couramment utilisée. Le concept sous-jacent reste cependant similaire : alors que le BIP-0038 vise à sécuriser une clé privée individuelle, le BIP-0039, lui, offre une protection à l'ensemble d'un portefeuille HD.

Termes associés : **BIP-0039** **CLÉ PRIVÉE**

BIP-0039

BIP

Le BIP-0039 introduit une méthode pour convertir la graine aléatoire d'un portefeuille en une suite de mots mémorisables et lisibles par l'Homme, connue sous le nom de phrase mnémonique. Cette phrase, généralement composée de 12 ou de 24 mots, permet de régénérer l'ensemble des clés privées d'un portefeuille de manière déterministe. Ainsi, au lieu de devoir mémoriser ou stocker une graine cryptographique complexe, les utilisateurs peuvent sauvegarder leurs bitcoins via une phrase de quelques mots. Le BIP-0039 a ainsi contribué à simplifier la gestion d'un portefeuille Bitcoin.

Terme associé : **PHRASE DE RÉCUPÉRATION**

BIP-0042

BIP

Proposition d'amélioration de Bitcoin Core qui adresse un bug mineur dans le calendrier de réduction de la récompense de bloc. Cette anomalie, si elle n'était pas corrigée, aurait conduit à une création totale de bitcoins supérieure à la limite prévue des 21 millions. Plus précisément, après la fin des *halvings*, un nouveau cycle de création de bitcoins aurait pu théoriquement se déclencher vers l'an 2214. Le code de Core en question utilisait une opération de décalage binaire à droite sur la valeur de la récompense du mineur. Le bug provenait de l'utilisation de cette opération de décalage dans un contexte où le comportement était non défini selon les normes du langage C++. Le décalage d'un entier de 64 bits (`int64_t`) de plus de 63 bits à droite entre dans cette catégorie de comportement non défini. Dans le code de Bitcoin Core, cela aurait pu se produire après 64 *halvings*, à la hauteur de bloc n° 13 440 000. Au-delà de cette limite, le comportement du décalage de bits n'était pas défini, ce qui signifie que différentes compilations pourraient interpréter le code différemment. Cela aurait pu entraîner des résultats imprévisibles, y compris la possibilité de créer une inflation infinie de bitcoins. Le BIP-0042 a corrigé ce problème en imposant que la récompense du bloc soit mise à zéro après 64 *halvings*, évitant ainsi l'utilisation d'une opération de décalage à droite dans un contexte de comportement non défini. Cette modification, qui était un *soft fork*, a ainsi permis de clarifier le comportement du code de Bitcoin Core. Bien que tout à fait sérieux, ce bug corrigé par le BIP-0042 n'était pas immédiatement critique, puisqu'il ne se serait manifesté qu'aux environs de l'an 2214. Publié le 1er avril 2014 par Pieter Wuille, le BIP-0042 se distingue ainsi par son ton humoristique.

Termes associés : **HALVING** **SOFT FORK**

BIP-0043

BIP

Proposition d'amélioration qui introduit l'utilisation d'un étage de dérivation pour décrire l'objectif (« *purpose field* ») dans la structure des portefeuilles HD, précédemment introduits dans le BIP-0032. Selon le BIP-0043, le premier niveau de dérivation d'un portefeuille HD, juste après la clé maîtresse notée `m/`, est réservé au numéro de l'objectif qui indique le standard de dérivation utilisé pour le reste du chemin. Ce numéro est enregistré comme un index (endurci). Par exemple, si le portefeuille suit le standard SegWit (BIP-0084), le début de son chemin de dérivation sera : `m/84'/`. Le BIP-0043 permet ainsi de clarifier les standards respectés par chaque logiciel de portefeuille pour avoir une meilleure interopérabilité entre eux. La standardisation de la suite du chemin de dérivation est décrite dans le BIP-0044.

Termes associés : **BIP-0032** **BIP-0044**

BIP-0044

BIP

Proposition d'amélioration qui introduit une structure de dérivation hiérarchique standard pour les portefeuilles HD. Le BIP-0044 s'appuie sur les principes établis par le BIP-0032 pour la dérivation des clés et sur le BIP-0043 pour l'utilisation du champ « *purpose* ». Il introduit une structure de cinq niveaux de dérivation : `m / purpose' / coin_type' / account' / change / address_index`. Voici le détail de chaque profondeur :

- `m /` indique la clé privée maîtresse. Elle est unique pour un portefeuille et ne peut pas avoir de sœurs à la même profondeur. La clé maîtresse est directement dérivée depuis la graine du portefeuille ;
- `m / purpose' /` indique l'objectif de dérivation qui permet d'identifier le standard suivi. Ce champ est décrit dans le BIP-0043. Par exemple, si le portefeuille respecte le standard BIP-0084 (SegWit V0), l'index sera alors `84'` ;
- `m / purpose' / coin-type' /` indique le type de cryptomonnaie. Cela permet de bien différencier les branches dédiées à une cryptomonnaie, des branches dédiées à une autre cryptomonnaie sur un portefeuille multi-coins. L'index dédié au Bitcoin est le `0'` ;
- `m / purpose' / coin-type' / account' /` indique le numéro de compte. Cette profondeur permet de différencier et d'organiser facilement un portefeuille en différents comptes. Ces comptes sont numérotés à partir de `0'`. Les clés étendues (`xpub`, `xprv`...) se trouvent à ce niveau de profondeur ;

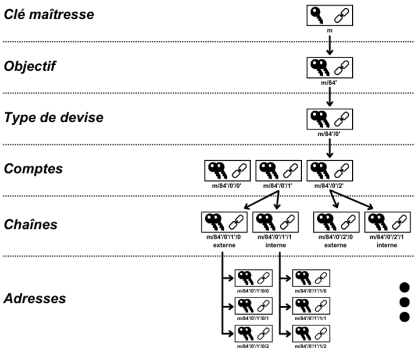
`m / purpose' / coin-type' / account' / change /` *indique la chaîne. Chaque compte tel que défini en profondeur 3 dispose de deux chaînes en profondeur 4 : une chaîne externe et une chaîne interne (également appelée «* change* »). La chaîne externe dérive des adresses destinées à être communiquées publiquement, c'est-à-dire les adresses que l'on nous propose lorsque l'on clique sur « recevoir » dans notre logiciel de portefeuille. La chaîne interne dérive les adresses destinées à ne pas être échangées publiquement, c'est-à-dire principalement les adresses de change. La chaîne externe est

identifiée avec l'index `0` et la chaîne interne est identifiée avec l'index `1`. Vous remarquerez qu'à partir de cette profondeur, on ne réalise plus une dérivation endurcie, mais une dérivation normale (il n'y a pas d'apostrophe). C'est grâce à ce mécanisme que l'on est capable de dériver l'ensemble des clés publiques enfants à partir de leur `xpub` ;

- `m / purpose' / coin-type' / account' / change / address-index` indique simplement le numéro de l'adresse de réception et de sa paire de clés, afin de la différencier de ses sœurs à la même profondeur sur la même branche. Par exemple, la première adresse dérivée dispose de l'index `0`, la deuxième adresse dispose de l'index `1`, etc...

Par exemple, si mon adresse de réception dispose du chemin de dérivation `m / 86' / 0' / 0' / 0 / 5`, on peut en déduire les informations suivantes :

- `86'` indique que nous suivons le standard de dérivation du BIP-0086 (Taproot ou SegWit V1) ;
- `0'` indique que c'est une adresse Bitcoin ;
- `0'` indique que l'on est sur le premier compte du portefeuille ;
- `0` indique que c'est une adresse externe ;
- `5` indique que c'est la sixième adresse externe de ce compte.



Termes associés : **BIP-0043** **CHEMIN DE DÉRIVATION**

BIP-0045

BIP

Proposition définissant un chemin de dérivation BIP-0032 spécifique pour les portefeuilles multisignatures P2SH de type déterministe. Le chemin utilisé est `m/45'/cosigner_index/change/address_index`, où chaque cosignataire se voit attribuer un index distinct. Cette structure permet à tous les participants de générer de manière déterministe les mêmes adresses multisig, sans nécessiter de communication supplémentaire après l'échange initial des clés publiques étendues. Le BIP-0045 a été largement supplanté par le BIP-0048 et le BIP-0087, qui offrent une meilleure prise en charge des différents types de scripts.

Termes associés : **BIP-0032** **MULTISIG**

BIP-0046

BIP

Définition d'un chemin de dérivation BIP-0032 spécifique pour les *fidelity bonds* timelockés, utilisés dans le protocole JoinMarket. Les fonds sont bloqués via `OP_CHECKLOCKTIMEVERIFY`, prouvant un sacrifice économique temporaire de la part du *maker*. Ce sacrifice augmente sa réputation et sa priorité dans le protocole *coinjoin*, car il démontre un engagement financier réel. Le BIP-0046 standardise le chemin de dérivation `m/84'/0'/0'/2/index`, où l'index (de 0 à 959) correspond à un mois entre janvier 2020 et décembre 2099, déterminant le *timelock* de l'adresse. Ce schéma permet de conserver les *fidelity bonds* en stockage à froid, car l'utilisateur n'a besoin de sauvegarder que sa graine et la référence à ce BIP, sans avoir à mémoriser les dates de verrouillage. Ce BIP est actuellement au statut de brouillon.

Termes associés : **BIP-0032** **JOINMARKET**

BIP-0047

BIP

Proposé par Justus Ranvier en 2015, ce protocole vise à résoudre le problème critique de la réutilisation des adresses Bitcoin, une pratique qui compromet gravement la confidentialité des utilisateurs sur le système. Satoshi Nakamoto, dans le White Paper de Bitcoin, avait déjà souligné l'importance d'utiliser des paires de clés distinctes pour chaque transaction afin de maintenir une ségrégation dans les différentes actions des utilisateurs. Le BIP-0047 introduit le concept de codes de paiement réutilisables, permettant à un utilisateur de recevoir de multiples paiements sans avoir à générer une nouvelle adresse Bitcoin manuellement pour chaque transaction. Ces codes agissent comme des identifiants virtuels, dérivés de la graine du portefeuille de l'utilisateur et situés au niveau des comptes dans la structure de dérivation d'un portefeuille HD. À partir de la combinaison des codes de paiements des 2 parties, chacune peut dériver un grand nombre d'adresses uniques appartenant à l'autre partie, sans avoir besoin de communiquer de nouveau avec elle. Le cœur de ce protocole repose sur l'algorithme ECDH (*Elliptic-Curve Diffie-Hellman*), une variante de l'échange de clés Diffie-Hellman établi sur les courbes elliptiques, qui permet aux deux parties d'établir un secret partagé pour la génération d'adresses de réception uniques. Bien que le BIP-0047 n'ait pas été ajouté à Bitcoin Core et ait reçu un accueil mitigé de la part de la communauté, des implémentations telles que PayNym sur Samourai Wallet et Sparrow Wallet l'ont adopté et l'ont pleinement intégré à leur écosystème d'outils de confidentialité.

Termes associés : **ECDH** **PAYNYM**

BIP-0048

BIP

Proposition définissant le chemin de dérivation pour les portefeuilles multisignatures prenant en charge plusieurs types de scripts :

```
m/48'/coin_type'/account'/script_type'
```

Le niveau `script_type'` permet de distinguer le format d'adresse utilisé : `1'` pour P2SH-P2WSH et `2'` pour P2WSH natif. Cette structure permet à un même portefeuille matériel de gérer plusieurs configurations multisig avec des formats de scripts différents, tout en maintenant une séparation claire entre chacune. Le BIP-0048 est largement adopté dans l'écosystème applicatif et matériel. Il constitue actuellement le standard de facto pour la dérivation de clés dans les portefeuilles multisignatures.

Termes associés : **BIP-0032** **MULTISIG**

BIP-0049

BIP

BIP de spécification qui introduit la méthode de dérivation utilisée pour générer des adresses Nested SegWit dans un portefeuille HD. Le chemin de dérivation proposé suit les standards du BIP-0043 et du BIP-0044, avec l'index `49'` (dérivation renforcée) à la profondeur de l'objectif. Par exemple, la première adresse d'un compte P2SH-P2WPKH serait issue du chemin : `m/49'/0'/0'/0/0`. Les scripts Nested SegWit ont été inventés au lancement de SegWit pour faciliter son adoption. Ils permettent d'utiliser ce nouveau standard, même sur des *wallets* pas encore compatibles nativement avec SegWit.

Terme associé : **P2SH-P2WPKH**

BIP-0050

BIP

BIP informatif faisant état d'un bug lié au passage de BerkeleyDB à LevelDB provoquant une division de la blockchain Bitcoin puis une réorganisation majeure de 24 blocs le 12 mars 2013. Ce BIP détaille l'incident et les actions correctives implémentées.

Termes associés : **BERKELEYDB** **LEVELDB**

BIP-0052

BIP

Proposition d'un nouveau paradigme de preuve de travail appelé « Optical Proof of Work » (oPoW), visant à réduire la consommation énergétique du minage de Bitcoin. Le BIP-0052 introduit « HeavyHash », une construction cryptographique qui combine un hachage Keccak (SHA-3) avec une multiplication matrice-vecteur. Cette opération de multiplication est conçue pour être exécutée de manière optimale par des coprocesseurs photoniques, une nouvelle classe de puces utilisant la lumière au lieu de courants électriques pour effectuer les calculs. L'objectif est de déplacer les coûts du minage des dépenses

opérationnelles (OPEX), principalement l'électricité, vers les dépenses en capital (CAPEX), c'est-à-dire le matériel. Cela permettrait de rendre le minage rentable indépendamment du coût local de l'électricité, favorisant ainsi une plus grande décentralisation géographique. Bien que le BIP soit classifié comme un hard fork, les auteurs suggèrent qu'il serait possible d'implémenter oPoW via un soft fork en *dual PoW* (SHA-256 et oPoW en parallèle). Cette proposition reste au stade de brouillon.

Termes associés : **PREUVE DE TRAVAIL** | **RÈGLES DE CONSENSUS**

BIP-0053

BIP

Proposition d'interdire les transactions de 64 octets afin de corriger une vulnérabilité dans la vérification des arbres de Merkle. Une transaction sérialisée de 64 octets peut être confondue avec un noeud interne de l'arbre de Merkle, car les deux partagent la même taille. Cette ambiguïté permettrait potentiellement de construire des preuves SPV frauduleuses trompant les nœuds légers. Le BIP-0053 fait partie du « consensus cleanup » proposé dans le BIP-0054, un ensemble de corrections mineures des règles de consensus.

Termes associés : **ARBRE DE MERKLE** | **RÈGLES DE CONSENSUS**

BIP-0054

BIP

Proposition actuellement en brouillon d'un ensemble de corrections de vulnérabilités dans les règles de consensus de Bitcoin, connue sous le nom de « Consensus Cleanup ». Le BIP-0054 corrige : *l'attaque* time warp* en imposant des contraintes sur les horodatages des blocs aux frontières des périodes d'ajustement de la difficulté,

- limite le temps de validation des blocs en plafonnant le nombre d'opérations de signature par transaction,
- invalide les transactions de 64 octets pour corriger les faiblesses de l'arbre de Merkle,
- et impose l'utilisation du champ `nLockTime` dans les transactions coinbase pour éliminer le besoin de validation BIP-0030.

Ce soft fork vise à réduire la surface d'attaque du protocole en corrigeant proprement des bugs historiques connus et documentés, sans modifier son fonctionnement de base.

Termes associés : **ARBRE DE MERKLE** | **RÈGLES DE CONSENSUS**

BIP-0060

BIP

Proposition visant à fixer la taille du message `version` du protocole P2P de Bitcoin et à y inclure proprement un champ booléen `relay`, indiquant si le nœud souhaite recevoir les transactions non confirmées relayées par ses pairs. Le BIP-0060 recommandait une approche à taille fixe pour garantir la rétrocompatibilité du *handshake* entre nœuds de versions différentes.

En pratique, le BIP-0037 (*Bloom filters*) avait déjà ajouté ce champ `relay` au message `version`, mais de manière non rétrocompatible : les anciens nœuds validant la taille du message échouaient au *handshake* avec les nœuds plus récents. De plus, l'implémentation était incohérente entre les versions du protocole (la version 70001 lisait `relay` sans l'écrire, la 70002 faisait les deux). Le BIP-0060 proposait de corriger cette situation, mais comme le BIP-0037 avait déjà cassé la compatibilité de manière irréversible, appliquer rétroactivement l'approche du BIP-0060 n'en valait plus l'effort. Son statut est donc « Closed ».

Termes associés : **BIP** | **RÉSEAU BITCOIN**

BIP-0061

BIP

Introduit un message de rejet dans le protocole de communication entre les nœuds. L'objectif du BIP-0061 est d'ajouter un mécanisme de retour d'information lorsqu'un nœud reçoit, de la part d'un autre nœud, une transaction ou un bloc qu'il considère comme invalide. Ce message de rejet permettrait à un nœud de signaler à l'expéditeur la raison pour laquelle cela a été rejeté. Ce type de communication devait contribuer à améliorer l'interopérabilité entre les différents clients et les communications entre les nœuds complets et les clients SPV. Les fonctionnalités amenées par le BIP-0061 ont finalement été abandonnées à partir de la version 0.20.0 de Bitcoin Core, car elles étaient considérées comme inutiles alors qu'elles impliquaient des besoins accrus en bande passante.

Termes associés : **NOEUD COMPLET** | **NOEUD SPV - NOEUD LÉGER**

BIP-0062

BIP

Proposition de plusieurs règles pour résoudre la malléabilité des transactions Bitcoin, incluant l'encodage DER strict des signatures, les *push data* canoniques et d'autres contraintes sur le format des scripts. Le BIP-0062 constitue un travail préparatoire important qui a influencé la conception de SegWit. Il a été retiré (*Withdrawn*), car ses règles étaient jugées insuffisantes pour couvrir tous les cas d'usage nécessitant la résolution de la malléabilité (comme les canaux de paiement) et trop complexes à implémenter en tant que règles de consensus. Le problème de malléabilité a finalement été résolu de manière plus complète par SegWit (BIP-0141), qui sépare les données de témoin du calcul du TXID.

Termes associés : **MALLÉABILITÉ - TRANSACTION** | **SEGWIT**

BIP-0064

BIP

Proposition d'un message P2P `getutxos` permettant d'interroger un nœud sur l'état d'un UTXO spécifique. Le BIP-0064 a été implémenté temporairement par Mike Hearn dans Bitcoin XT, puis retiré en raison de problèmes de sécurité, car il facilitait les attaques par déni de service et permettait de fournir de fausses informations aux clients légers.

Termes associés : **RÉSEAU BITCOIN** | **UTXO**

BIP-0065

BIP

Introduit un nouvel opcode nommé `OP_CHECKLOCKTIMEVERIFY` qui permet de rendre un UTXO inutilisable jusqu'à un moment donné dans le futur. L'application de ce BIP a nécessité un soft fork, qui est intervenu le 14 décembre 2015. Il a également introduit la version 4 des blocs.

Termes associés : **OP_CHECKLOCKTIMEVERIFY - 0XB1** | **TIMELOCK**

BIP-0066

BIP

Introduit une standardisation du format des signatures dans les transactions. Ce BIP a été proposé en réaction à une divergence dans la manière dont OpenSSL gérait l'encodage des signatures sur différents systèmes. Cette hétérogénéité posait un risque de scission de la blockchain. Le BIP-0066 a permis d'uniformiser le format des signatures pour toutes les transactions en utilisant l'encodage DER strict (*Distinguished Encoding Rules*). Cette modification nécessitait un soft fork. Pour son activation, le BIP-0066 a alors utilisé le même mécanisme que le BIP-0034, nécessitant l'augmentation du champ `nVersion` à sa version 3, et rejetant tous les blocs de version 2 ou inférieure une fois que le seuil de 95 % des mineurs était atteint. Ce seuil a été franchi au bloc n° 363 725 le 4 juillet 2015.

Terme associé : **DER**

BIP-0067

BIP

Proposition spécifiant que les clés publiques au sein d'un `redeemScript` multisignature P2SH doivent être triées dans l'ordre lexicographique. Ce tri garantit que tous les cosignataires génèrent la même adresse multisig, indépendamment de l'ordre dans lequel les clés sont insérées lors de la création du script. Sans cette norme, des ordres de clés différents produiraient des `redeemScript` et donc des adresses distincts, empêchant l'interopérabilité entre logiciels de portefeuilles.

Termes associés : **MULTISIG** | **P2SH**

BIP-0068

BIP

Introduit la possibilité d'utiliser des blocages temporels relatifs (*relative locktime*) grâce au champ `nSequence`. Cela permet à une transaction de spécifier un délai relatif avant qu'elle soit incluse dans un bloc. Ce délai peut être défini en termes de nombre de blocs, ou bien comme un multiple de 512 secondes (c'est-à-dire, du temps réel). Notons que cette nouvelle interprétation du champ `nSequence` est uniquement valide si le champ `nVersion` est supérieur ou égal à `2`. Cette interprétation du champ `nSequence` se fait au niveau des règles de consensus de Bitcoin. Le timelock relatif définit un délai à partir de l'acceptation d'une transaction antérieure alors que le timelock absolu spécifie un moment précis avant lequel la transaction ne peut être incluse dans un bloc. Le BIP-0068 a été introduit via un soft fork le 4 juillet 2016 en même temps que le BIP-0112 et le BIP-0113, activé pour la première fois grâce à la méthode du BIP-0009.

Terme associé : **NSEQUENCE**

BIP-0069

BIP

Proposition suggérant un tri lexicographique des entrées et des sorties d'une transaction Bitcoin afin de réduire les possibilités d'identification par empreinte (*fingerprinting*). Les entrées sont triées par `txid` puis par index de sortie (`vout`), et les sorties par montant puis par `scriptPubKey`. L'objectif est d'améliorer la confidentialité en rendant les transactions produites par différents logiciels plus uniformes, ce qui complique l'identification du portefeuille utilisé. Toutefois, ce BIP est aujourd'hui considéré comme insuffisant en matière de confidentialité, car un tri déterministe peut lui-même devenir une empreinte identifiable.

Termes associés : **FRAIS DE TRANSACTION** **MEMPOOL**

BIP-0070

BIP

Protocole de paiement interactif pour Bitcoin. Il permet l'envoi de demandes de paiement et la réception de paiements de manière sécurisée et standardisée. Dans ce protocole, le client clique sur une URI Bitcoin (BIP-0021) étendue avec un paramètre supplémentaire (décrit dans le BIP-0072). La demande de paiement est signée avec le certificat SSL du commerçant. Lors de la réception et de la validation de cette demande, les détails du paiement sont automatiquement remplis dans l'interface de la transaction du portefeuille du client. Ce protocole fournit une confirmation de paiement et améliore la sécurité et l'expérience utilisateur en clarifiant l'entité bénéficiaire du paiement. Cette méthode proposée dans le BIP-0070 n'a jamais été largement adoptée par les commerçants.

Termes associés : **BIP-0021** **BIP-0072**

BIP-0071

BIP

Définit un type de média MIME (*Multipurpose Internet Mail Extensions*), conformément à la norme RFC 2046, pour les messages de demande de paiement en bitcoins dans le BIP-0070. MIME est un standard Internet qui étend le format des messages électroniques pour permettre l'envoi de divers types de données de manière structurée. Dans le BIP-0071, l'adoption d'un type MIME spécifique pour les messages de paiement assure que les logiciels de portefeuille, lorsqu'ils envoient des messages de protocole de paiement par e-mail ou HTTP, respectent les standards Internet pour une encapsulation appropriée des messages. Cette amélioration étant groupée avec le BIP-0070, elle n'a jamais été largement adoptée.

Termes associés : **BIP-0070** **BIP-0072**

BIP-0072

BIP

Complète le BIP-0070 et le BIP-0071 en définissant l'extension des URI Bitcoin (BIP-0021) avec un paramètre supplémentaire `r`. Ce paramètre permet d'inclure un lien vers une demande de paiement sécurisée et signée par le certificat SSL du commerçant. Lorsqu'un client clique sur cette URI étendue, son portefeuille contacte le serveur du commerçant pour demander les détails de paiement. Ces détails sont automatiquement remplis dans l'interface de transaction du portefeuille et le client peut être informé qu'il paie le propriétaire du domaine correspondant au certificat de signature (par exemple, « pandul.fr »). Cette amélioration étant groupée avec le BIP-0070, elle n'a jamais été largement adoptée.

Termes associés : **BIP-0021** **BIP-0070**

BIP-0073

BIP

Extension du protocole de paiement BIP-0070 utilisant l'en-tête HTTP `Accept` pour la négociation du format de réponse entre le client et le serveur marchand. Le BIP-0073 permettait au portefeuille de spécifier les types de contenu acceptés. Déployé conjointement avec le BIP-0070, il est devenu obsolète avec l'abandon général du protocole de paiement.

Termes associés : **BIP-0021** **BIP-0070**

BIP-0074

BIP

Extension du protocole de paiement BIP-0070 permettant d'inclure des sorties `OP_RETURN` à valeur nulle dans les requêtes de paiement. Le BIP-0074 visait à autoriser l'intégration de données arbitraires (métadonnées, identifiants de commande, preuves) dans les transactions générées par le protocole de paiement, sans détruire de fonds dans ces sorties. Il a été rejeté de manière unanime par la communauté (*Unanimously Discourage for implementation*), son statut officiel étant *Rejected*.

Termes associés : **BIP-0070** **OP_RETURN - 0X6A**

BIP-0075

BIP

Extension qui améliore le protocole de paiement BIP-0070 en introduisant deux innovations majeures. Premièrement, il permet à l'expéditeur d'une demande de paiement de signer volontairement cette requête et de fournir un certificat qui permet au destinataire de connaître l'identité de son interlocuteur. Secondement, il chiffre la demande de paiement retournée pour prévenir toute interception et visualisation par des tiers. Cela permet de renforcer la sécurité et la confidentialité des transactions en assurant que les détails du paiement ne sont visibles que par les participants. Le BIP-0075 offre aussi de nouvelles fonctionnalités pour améliorer l'expérience utilisateur. Cette amélioration étant établie sur le BIP-0070, elle n'a jamais été largement adoptée.

Termes associés : **BIP-0070** **BIP-0072**

BIP-0077

BIP

Proposition d'une version asynchrone du protocole de Payjoin (BIP-0078). Le BIP-0077 permet au récepteur d'être hors ligne lors de l'initiation du paiement. Le payeur publie une transaction PSBT chiffrée sur un serveur relais, que le récepteur récupère et co-signe ultérieurement. Cette approche améliore considérablement l'utilisabilité du Payjoin original, qui nécessitait que les deux parties soient en ligne simultanément pour compléter la transaction collaborative. Le protocole maintient les mêmes avantages en termes de confidentialité.

Termes associés : **BIP-0078** **PAYJOIN**

BIP-0078

BIP

Introduit un protocole pour utiliser des Payjoin sur Bitcoin, une structure de transaction qui améliore la confidentialité des paiements en faisant intervenir le destinataire auprès du payeur dans les inputs. Ce BIP s'inspire du BIP-0079, qui avait déjà présenté un concept semblable. Toutefois, ces deux BIP ne marquent pas l'origine du concept de Payjoin. En effet, l'implémentation Stowaway de Samourai Wallet existait déjà auparavant et cette structure de transaction avait été mentionnée pour la première fois par LaurentMT en 2015, sous le nom de « *steganographic transaction* ».

Terme associé : **PAYJOIN**

BIP-0079

BIP

Proposition antérieure au BIP-0078 pour un protocole de « *pay-to-endpoint* » (P2EP), également connu sous le nom de Bustapay. Le récepteur ajoutait un de ses inputs à la transaction du payeur, cassant ainsi l'heuristique d'analyse de chaîne qui suppose que tous les inputs appartiennent au même propriétaire. Le BIP-0079 a été supplanté par le Payjoin (BIP-0078) qui a formalisé et amélioré cette approche de transaction collaborative.

Termes associés : **COINJOIN** **PAYJOIN**

BIP-0080

BIP

Proposition fermée définissant un chemin de dérivation hiérarchique déterministe (BIP-0032) pour des portefeuilles multisig déterministes utilisés dans des « *voting pools* » sans *colored coins*. Un « *voting pool* » est un mécanisme de gestion collaborative de fonds en *cold storage* de type FIFO, où plusieurs participants partagent le contrôle de portefeuilles multisig. Le BIP-0080, application spécifique du BIP-0043 basée sur le BIP-0044, définit le chemin `m / purpose' / coin_type' / series' / address_index` permettant à chaque participant de dériver des clés de manière coordonnée pour construire les scripts de dépôt multisig. Cette proposition très spécialisée n'a jamais été largement adoptée par l'écosystème Bitcoin.

Termes associés : **BIP-0032** **MULTISIG**

BIP-0081

BIP

Variante du BIP-0080 définissant un chemin de dérivation hiérarchique déterministe (BIP-0032) pour les pools de vote utilisant des colored coins avec des portefeuilles multisig déterministes. Le BIP-0081 étendait le schéma du BIP-0080 aux actifs tokenisés en ajoutant cinq niveaux de dérivation supplémentaires pour encoder la définition de couleur (le hash160 de la définition, divisé en cinq groupes de 4 octets en little-endian) au sein de la pool. Comme le BIP-0080, cette proposition est restée très spécialisée et n'a jamais été adoptée par l'écosystème Bitcoin.

Termes associés : **BIP-0032** **COLORED COINS**

BIP-0083

BIP

Proposition d'une extension des arbres de clés hiérarchiques déterministes (BIP-0032) permettant la création dynamique de nouveaux niveaux de dérivation selon les besoins de l'application. Le BIP-0083 a été fermé, les chemins de dérivation standards définis par les BIP-0044, BIP-0049 et BIP-0084 ayant été jugés suffisants pour la majorité des cas d'utilisation.

Termes associés : **BIP** **BIP-0032**

BIP-0084

BIP

Définit le standard de dérivation des adresses SegWit V0 (`bc1q...`) au sein d'un portefeuille déterministe et hiérarchique. Il définit l'index `84'` qui doit désormais être utilisé à la profondeur `purpose` du portefeuille HD pour les modèles de script P2WPKH.

Termes associés : **BIP-0032** **BIP-0043** **P2WPKH**

BIP-0085

BIP

Solution pour unifier la dérivation de différents portefeuilles Bitcoin en utilisant une graine maîtresse unique pour tous. Cette proposition permet de dériver de l'entropie à partir d'une information racine pour générer plusieurs phrases mnémoniques pour plusieurs portefeuilles, sans compromettre la sécurité. L'objectif du BIP-0085 est de faciliter la gestion et la sauvegarde de plusieurs portefeuilles Bitcoin. Au lieu de devoir sécuriser plusieurs phrases, une seule information suffit pour toutes les autres.

Termes associés : **ENTROPIE** **GRAINE**

BIP-0086

BIP

Définit le standard de dérivation des adresses SegWit V1 ou Taproot (`bc1p …`) au sein d'un portefeuille déterministe et hiérarchique. Il définit l'index `86'` qui doit désormais être utilisé à la profondeur `purpose` du portefeuille HD pour les modèles de script P2TR.

Termes associés : **BIP-0032** | **BIP-0043** | **P2TR**

BIP-0087

BIP

Proposition définissant le chemin de dérivation `m/87'/coin_type'/account'` spécifiquement pour les portefeuilles multisignatures utilisant les *output script descriptors*. Contrairement au BIP-0048 qui encode le type de script dans le chemin de dérivation, le BIP-0087 adopte une approche plus simple en déléguant la spécification du type de script au descriptor lui-même. Cette conception est mieux adaptée à l'écosystème moderne des portefeuilles qui s'appuient sur les descriptors pour décrire précisément les conditions de dépense.

Termes associés : **BIP-0032** | **MULTISIG**

BIP-0088

BIP

Proposition définissant un format de template pour les chemins de dérivation BIP-0032. Ce format permet de spécifier des plages et des caractères génériques (*wildcards*) au sein des chemins, par exemple `m/49'/0'/0'/{0,1}/*` pour indiquer que le niveau de change peut être 0 ou 1, et que l'index d'adresse est libre. Ces templates permettent aux portefeuilles de décrire de manière formelle quels chemins de dérivation sont autorisés, facilitant ainsi la validation et la vérification des politiques de portefeuille.

Termes associés : **BIP-0032** | **CHEMIN DE DÉRIVATION**

BIP-0089

BIP

Proposition (brouillon) décrivant la *Chain Code Delegation* (CCD), un mécanisme pour les portefeuilles multisignatures dans lequel un participant privilégié (le « délégataire ») retient les codes de chaîne BIP-0032 d'un ou de plusieurs participants non privilégiés (les « délégateurs »), et leur fournit des ajustements scalaires (*tweaks*) par entrée au moment de la signature. Les délégateurs ne détiennent qu'une simple paire de clés (sans clé étendue) et ne peuvent donc ni dériver d'autres adresses du portefeuille, ni observer le solde ou l'activité de dépense des autres combinaisons de cosignataires. Ce BIP est particulièrement utile dans les configurations de garde collaborative où le propriétaire du portefeuille souhaite préserver la confidentialité de son solde vis-à-vis de son cosignataire ou *custodian*. Le BIP-0089 prend en charge deux modes de signature : un mode non aveuglé, où le délégateur voit la transaction qu'il signe, et un mode aveuglé (*blinded*), où il ne voit ni le message ni la clé publique enfant

utilisée.

Terme associé : **BIP-0032**

BIP-0090

BIP

Proposition pour simplifier le traitement de l'activation des soft forks antérieurs en remplaçant le mécanisme de signalisation par les mineurs via les numéros de version des blocs par de simples vérifications de la hauteur de bloc. Cette modification éliminerait la nécessité de vérifier les 1000 blocs précédents pour l'activation des règles de consensus, ce qui permettrait de réduire la dette technique associée à l'implémentation de ces soft fork.

Termes associés : **BIP-0009** **SOFT FORK**

BIP-0091

BIP

Proposition de James Hilliard pour faciliter l'activation du soft fork SegWit, défini dans les BIP-0141, BIP-0143 et BIP-0147, via un MASF sans atteindre directement le seuil requis de 95 % de la puissance de calcul signalant le soutien via le bit 1. Le BIP-0091 permet aux mineurs de signaler indirectement leur soutien à SegWit en utilisant le bit 4 dans les blocs minés. Une fois que 269 blocs sur une fenêtre de 336 blocs ont inclus le bit 4 (soit 80 % de la puissance de calcul), le BIP-0091 se verrouille, obligeant ensuite tous les nœuds compatibles à rejeter les blocs n'incluant pas le bit 1. Cette méthode visait à rendre le BIP-0148 (UASF) obsolète et à éviter une scission potentielle de la blockchain le 1er août 2017. Le BIP-0091 a finalement été activé le 23 juillet 2017 (au bloc 477 120), juste avant la date fatidique du 1er août imposée dans le BIP-0148. Cela a permis de forcer le signalement de SegWit par les mineurs, qui sera finalement verrouillé le 9 août au bloc 479 808, puis activé le 24 août au bloc 481 824. Pour résumer, le BIP-0148 (UASF) a été créé en réaction au fait que les mineurs ne signalaient pas suffisamment SegWit, mais n'a finalement jamais été mis en œuvre. Le BIP-0091 (MASF) a été créé en réaction au BIP-0148 afin de forcer la main aux mineurs, sans pour autant risquer l'UASF du BIP-0148. Le BIP-0091 représente lui-même un soft fork, qui forcera finalement les mineurs à verrouiller le soft fork SegWit via la méthode de base (MASF BIP-0009).

Termes associés : **BIP-0148** **MASF**

BIP-0093

BIP

BIP informationnel qui suggère un standard pour sauvegarder et restaurer la graine d'un portefeuille déterministe hiérarchique (selon le BIP-0032) en utilisant le Partage de secret de Shamir. Ce protocole définit le format « codex32 », qui s'inspire du format bech32, en introduisant une chaîne structurée composée d'un préfixe, d'un paramètre de seuil, d'un identifiant, d'un index de partage, d'une charge utile et d'une checksum (BCH). La méthode permet de scinder la graine en jusqu'à 31 parts, dont un seuil défini (entre 2 et 9) est requis pour la récupération complète de la graine.

Termes associés : **BIP-0032** **GRAINE**

BIP-0094

BIP

Définit le Testnet 4, un nouveau réseau testnet destiné à remplacer le Testnet 3 en 2024. Ce dernier, qui était en service depuis 13 ans, était devenu vulnérable à diverses attaques et avait été le théâtre d'abus d'utilisation par le biais d'airdrops frauduleux d'altcoins, perturbant la distribution normalement gratuite de TBTC. Pour corriger ces failles, Testnet 4 intègre 3 ajustements majeurs dans les règles de consensus : la modification de l'exception des 20 minutes, une correction de l'algorithme d'ajustement de la difficulté (*block storm fix*) et la mise en place d'une protection contre l'attaque « time warp ».

Terme associé : **TESTNET**

BIP-0098

BIP

Proposition d'une variante optimisée des arbres de Merkle pour Bitcoin, offrant des preuves d'inclusion plus rapides et plus compactes. Le BIP-0098 a été initialement conçu dans le cadre des premières propositions de MAST (Merkelized Abstract Syntax Trees), visant à permettre l'exécution conditionnelle de branches de scripts. Ce travail préparatoire a influencé les développements ultérieurs qui ont abouti à Taproot (BIP-0341).

Termes associés : **ARBRE DE MERKLE** **SCRIPT**

BIP-0099

BIP

Document informatif discutant des motivations et des méthodes de déploiement des changements de règles de consensus sur Bitcoin. Le BIP-0099 analyse les risques associés aux hard forks et aux soft forks, et propose un cadre de réflexion pour évaluer les propositions de modification du protocole. Ce document n'introduit aucune modification technique et a été fermé.

Termes associés : **HARD FORK** **SOFT FORK**

BIP-0100

BIP

Proposition de Jeff Garzik durant la Blocksize War permettant aux mineurs de voter pour ajuster dynamiquement la taille maximale des blocs entre 1 et 32 Mo. Le mécanisme reposait sur un vote par bloc, où les mineurs exprimaient leur préférence pour une augmentation ou une diminution de la limite. Ce BIP-0100 a été rejeté, la communauté ayant finalement opté pour SegWit plutôt que pour une augmentation directe de la taille des blocs.

Termes associés : **BLOCKSIZE WAR** | **MAX_BLOC_SIZE**

BIP-0101

BIP

Proposition présentée par Gavin Andresen en 2015 qui visait à augmenter la taille maximale des blocs de 1 Mo à 8 Mo et à doubler cette limite tous les deux ans jusqu'à atteindre un peu plus de 8 Go en 2036. Cette proposition a été intégrée à l'implémentation Bitcoin XT en août 2015, qui est devenu de fait un fork de Bitcoin, mais n'a pas été suivie. Cette proposition n'a donc jamais été adoptée. Certains considèrent l'implémentation du BIP-0101 sur Bitcoin XT comme le casus belli de la Blocksize War.

Terme associé : **BITCOIN XT**

BIP-0102

BIP

Proposition présentée par Jeff Garzik en juin 2015 dans le cadre de la Blocksize War. Elle visait à augmenter la taille maximale des blocs de 1 Mo à 2 Mo par un hard fork. C'est cette proposition qui a par la suite inspiré le BIP-0109 et la mise en place de Bitcoin Classic. Cette proposition n'a jamais été adoptée.

Terme associé : **BITCOIN CLASSIC**

BIP-0103

BIP

Proposition de Pieter Wuille durant la Blocksize War prévoyant une augmentation progressive de la taille maximale des blocs de 17,7 % par an, suivant la croissance empirique observée de la bande passante. Approche conservatrice comparée aux autres propositions de l'époque, elle visait à maintenir la décentralisation du réseau. Le BIP-0103 n'a jamais été déployé, SegWit ayant été préféré comme solution au problème de scalabilité.

Termes associés : **BLOCKSIZE WAR** | **MAX_BLOC_SIZE**

BIP-0104

BIP

Proposition surnommée « Block75 » qui prévoyait un ajustement automatique de la taille maximale des blocs visant un taux de remplissage de 75 %. Le mécanisme était similaire au retargeting de la difficulté : si les blocs étaient trop pleins, la limite augmentait ; s'ils étaient trop vides, elle diminuait. Ce BIP-0104, proposé durant la Blocksize War, n'a jamais été adopté.

Termes associés : **AJUSTEMENT DE LA DIFFICULTE** | **BLOCKSIZE WAR**

BIP-0105

BIP

Proposition d'un algorithme de retargeting de la taille maximale des blocs basé sur le consensus des mineurs. Le BIP-0105 incluait un plafond de croissance pour éviter une augmentation excessive de la taille des blocs. Proposé durant la Blocksize War, il fait partie des nombreuses tentatives d'ajustement dynamique de la limite de taille des blocs qui n'ont jamais été déployées sur Bitcoin.

Termes associés : **BLOCKSIZE WAR** | **MAX_BLOC_SIZE**

BIP-0106

BIP

Proposition d'un contrôle dynamique de la taille maximale des blocs basé sur l'historique de remplissage des blocs précédents. Le BIP-0106 ajustait automatiquement la limite en fonction de la demande réelle d'espace dans les blocs. Proposé durant la Blocksize War, il n'a jamais été adopté, la communauté ayant finalement résolu le problème de scalabilité par l'activation de SegWit.

Termes associés : **BLOCKSIZE WAR** | **MAX_BLOC_SIZE**

BIP-0107

BIP

Proposition d'une limite dynamique de la taille des blocs en deux phases. La première phase prévoyait des augmentations programmées : 2 Mo en 2016-2017, 4 Mo en 2018-2019, puis 6 Mo en 2020. La seconde phase introduisait un mécanisme d'ajustement automatique : tous les 4 032 blocs (environ 4 semaines), si plus de 75 % des blocs étaient remplis à au moins 60 %, la taille maximale augmentait de 10 %. Ce mécanisme ne permettait que des augmentations, jamais de diminution. Le BIP-0107 a le statut « Closed » et n'a jamais été déployé sur Bitcoin.

Termes associés : **BLOCKSIZE WAR** | **MAX_BLOC_SIZE**

BIP-0109

BIP

Proposition présentée par Gavin Andresen en janvier 2016 dans le cadre de la Blocksize War. Elle visait à augmenter la taille maximale des blocs de 1 Mo à 2 Mo par un hard fork, tout en ajoutant certaines limites sur le nombre de sigops et de sighashs (contrairement au BIP-0102). L'activation du BIP-0109 nécessitait un soutien de 75 % de la puissance de hachage sur une fenêtre de 1 000 blocs, suivi d'une période de grâce de 28 jours. Cette proposition faisait partie intégrante du fork Bitcoin Classic, qui cherchait à offrir une solution pour le passage à l'échelle de Bitcoin. Cette proposition n'a jamais été adoptée.

Terme associé : **BITCOIN CLASSIC**

BIP-0110

BIP

Proposition de soft fork temporaire (un an) visant à limiter au niveau du consensus la taille des données arbitraires pouvant être inscrites dans les transactions Bitcoin, afin de lutter contre l'utilisation du système comme support de stockage de données (notamment les inscriptions Ordinals apparues en 2022).

Concrètement, le BIP-0110 invalide les `scriptPubKey` de plus de 34 octets (sauf `OP_RETURN` limité à 83 octets), les éléments de witness et les payloads `OP_PUSHDATA` dépassant 256 octets, les dépenses de versions witness non définies, l'annexe Taproot, les *control blocks* Taproot de plus de 257 octets (limitant l'arbre à 128 feuilles), ainsi que les opcodes `OP_SUCCESS` et `OP_IF`/`OP_NOTIF` dans Tapscript. Les UTXO créés avant l'activation sont exemptés.

Le déploiement utilise un BIP-0009 modifié avec un seuil réduit à 55 %, une activation obligatoire par hauteur de bloc, et une durée active de 52 416 blocs (environ 1 an), après quoi les règles expirent automatiquement. Ce BIP est l'un des plus controversés de l'ère post-Ordinals, ses partisans y voyant une défense de Bitcoin comme monnaie, ses détracteurs dénonçant une restriction arbitraire de l'espace de bloc.

Termes associés : **RÈGLES DE CONSENSUS** **SOFT FORK**

BIP-0111

BIP

Propose l'ajout d'un bit de service nommé `NODE_BLOOM` pour permettre aux nœuds de signaler explicitement leur prise en charge des Bloom Filters tels que décrits dans le BIP-0037. L'introduction de `NODE_BLOOM` permet aux opérateurs de nœuds de désactiver ce service afin de réduire les risques de DoS. L'option du BIP-0037 est par défaut désactivée dans Bitcoin Core. Pour l'activer, il faut entrer le paramètre `peerbloomfilters=1` dans le fichier de configuration.

Terme associé : **DOS - DENIAL OF SERVICE**

BIP-0112

BIP

Introduit l'opcode `OP_CHECKSEQUENCEVERIFY` (CSV) dans le langage Script de Bitcoin. Cette opération permet de créer des transactions dont la validité ne devient effective qu'après un certain délai relatif à une transaction antérieure, défini soit en nombre de blocs, soit en durée de temps. `OP_CHECKSEQUENCEVERIFY` compare la valeur en haut de la pile avec la valeur du champ `nSequence` de l'input. Si elle est supérieure et que toutes les autres conditions sont respectées, le script est valide. Ainsi, `OP_CHECKSEQUENCEVERIFY` restreint les valeurs possibles pour le champ `nSequence` de l'input qui le dépense, et ce champ `nSequence` restreint lui-même le moment où la transaction qui comprend cet input peut être incluse dans un bloc. Le BIP-0112 a été introduit via un soft fork le 4 juillet 2016 en même temps que le BIP-0068 et le BIP-0113, activé pour la première fois grâce à la méthode du BIP-0009.

Termes associés : **BIP-0068** **NSEQUENCE**

BIP-0113

BIP

A introduit une modification dans le calcul des opérations de timelock basées sur `nLockTime` et `OP_CHECKLOCKTIMEVERIFY`. Il spécifie que pour évaluer la validité de ces timelocks, il faut désormais les comparer au MTP (*Median Time Past*), c'est-à-dire la médiane des horodatages des 11 derniers blocs. Auparavant, on utilisait seulement l'horodatage du bloc courant. Cette méthode rend le système plus prévisible et évite la manipulation du référentiel de temps par les mineurs. Le BIP-0113 a été introduit via un soft fork le 4 juillet 2016 en même temps que le BIP-0068 et le BIP-0112, activé pour la première fois grâce à la méthode du BIP-0009.

Termes associés : **BIP-0112** **TIMELOCK**

BIP-0114

BIP

Première proposition formelle de MAST (*Merkelized Abstract Syntax Tree*) pour Bitcoin, présentée par Johnson Lau en 2016. Le BIP-0114 permettait de structurer les conditions de dépense d'un UTXO sous la forme d'un arbre de Merkle, où chaque feuille représente un script alternatif. Lors de la dépense, seule la branche exécutée devait être révélée, les autres restant masquées. Cela améliorait la confidentialité et réduisait la taille des transactions complexes. Ce BIP a été fermé au profit du BIP-0341 (Taproot), qui intègre le concept de MAST de manière plus élégante via son *script-path*.

Termes associés : **BIP-0341** **MAST**

BIP-0115

BIP

Proposition d'un mécanisme de protection contre le rejeu de transactions en cas de *hard fork*, en engageant chaque transaction sur un *blockhash* spécifique. Le BIP-0115 ajoutait un opcode permettant de vérifier qu'un bloc particulier existe dans la chaîne active, rendant la transaction invalide sur une chaîne alternative issue d'un fork. Cette proposition n'a jamais été déployée.

Termes associés : **REPLAY ATTACK** **SCRIPT**

BIP-0116

BIP

Propose l'ajout de l'opcode `OP_MERKLEBRANCHVERIFY` au système de script de Bitcoin. Cet opcode permet de vérifier une preuve d'inclusion dans un arbre de Merkle directement dans un script. Concrètement, il prend sur la pile un hash racine, une preuve de branche et une valeur, puis vérifie que cette valeur appartient bien à l'arbre identifié par la racine. Combiné avec d'autres opcodes, il permettrait de construire des structures de type MAST (*Merkelized Abstract Syntax Tree*) directement en Script. Ce BIP, proposé par Mark Friedenbach en 2017, a été supplanté par Taproot (BIP-0341) qui intègre nativement le concept de MAST.

Termes associés : **MAST** **OPCODES**

BIP-0117

BIP

Propose l'introduction d'une sémantique d'exécution par appel terminal (*tail call*) dans le système de Script de Bitcoin. Ce mécanisme permettrait à un script de transférer l'exécution vers un autre script sans augmenter la profondeur de la pile, de manière analogue à l'optimisation *tail call* dans les langages de programmation. Combiné avec le BIP-0116 (`OP_MERKLEBRANCHVERIFY`), il offrirait des capacités générales de MAST, permettant de sélectionner et d'exécuter dynamiquement un sous-script parmi un ensemble structuré en arbre de Merkle. Ce BIP, proposé par Mark Friedenbach en 2017, a été supplanté par Taproot (BIP-0341).

Termes associés : **OPCODES** **SCRIPT**

BIP-0118

BIP

Proposition d'amélioration de Bitcoin visant à introduire deux nouveaux SigHash Flag modificateurs : `SIGHASH_ANYPREVOUT` et `SIGHASH_ANYPREVOUTANYSCRIPT`. Ces fonctionnalités étendent les capacités des transactions Bitcoin, en particulier en ce qui concerne les contrats intelligents et les solutions de surcouches comme le Lightning Network. Le BIP-0118 permettrait notamment l'utilisation d'Eltoo. Le principal avantage du `SIGHASH_ANYPREVOUT` est de permettre la réutilisation des signatures dans plusieurs transactions, ce qui offre plus de flexibilité. Concrètement, ces SigHash permettent d'obtenir une signature qui ne couvre aucun input de la transaction.

Terme associé : **SIGHASH FLAG**

BIP-0119

BIP

Introduit un nouvel opcode nommé `OP_CHECKTEMPLATEVERIFY` (CTV). CTV permettrait de créer des covenants non récursifs dans les transactions, afin d'imposer des conditions spécifiques sur la manière dont une pièce donnée peut être dépensée, y compris dans des transactions futures. Plus concrètement, il permettrait de définir des conditions sur le `scriptPubKey` des sorties d'une transaction à partir du `scriptPubKey` de l'UTXO dépensé en entrée. CheckTemplateVerify est conçu pour être simple et sans état dynamique. Sa mise en œuvre vise à étendre les capacités de script de Bitcoin pour faciliter diverses applications telles que le contrôle de congestion des transactions, la création de canaux de paiement non interactifs, les DLC, les pools de paiement... Ce nouvel opcode serait introduit en remplacement de l'`OP_NOP4`. Cette modification impliquerait un soft fork.

Terme associé : **COVENANT**

BIP-0120

BIP

Proposition d'un standard de « preuve de paiement » permettant de prouver qu'une transaction Bitcoin a bien été envoyée, en signant les inputs de la transaction originale avec les mêmes clés. Le BIP-0120 visait à fournir une preuve cryptographique vérifiable que le payeur contrôlait les fonds utilisés dans une transaction spécifique. Il n'a jamais été largement adopté.

Termes associés : **BIP** **BIP-0070**

BIP-0121

BIP

Définition d'un schéma d'URI complémentaire au BIP-0120 pour les preuves de paiement, permettant de déclencher la génération et la vérification de preuves via un lien standardisé. L'URI contenait les paramètres nécessaires pour identifier la transaction originale et initier le protocole de preuve. Le BIP-0121 a été fermé en même temps que le BIP-0120, ces deux propositions complémentaires n'ayant jamais été adoptées par l'écosystème, les cas d'usage étant couverts par d'autres mécanismes.

Termes associés : **BIP-0021** **BIP-0120**

BIP-0122

BIP

Proposition d'un schéma d'URI pour référencer des éléments spécifiques de la blockchain (blocs, transactions) de manière standardisée. Le BIP-0122 faciliterait l'interopérabilité entre les explorateurs de blocs et les applications en définissant un format uniforme pour identifier une blockchain particulière et pointer vers ses données. Chaque blockchain est identifiée par le hash de son bloc de genèse. Ce BIP est resté au stade de brouillon depuis 2015.

Termes associés : **BIP** **BIP-0021**

BIP-0123

BIP

Établit un nouveau processus standardisé pour la classification des propositions d'amélioration de Bitcoin. Les BIP doivent dorénavant être classifiés selon 4 couches (*layers*) :

- `Consensus` : concerne les propositions qui nécessitent un changement de consensus et affectent la compatibilité entre les versions antérieures et futures du protocole Bitcoin. Ce sont les soft forks et les hard forks ;
- `Peer Services` : concerne les modifications des services et des protocoles de communication entre les nœuds du réseau, sans affecter le consensus ;
- `API/RPC` : englobe les propositions visant à modifier les API et les RPC utilisés pour interagir avec les nœuds Bitcoin ;
- `Applications` : comprend les propositions d'améliorations pour les applications qui s'exécutent au-dessus du réseau Bitcoin, comme typiquement les standards liés aux logiciels de portefeuilles.

Termes associés : **HARD FORK** **SOFT FORK**

BIP-0124

BIP

Proposition de templates de scripts hiérarchiques déterministes étendant le BIP-0032. Le BIP-0124 visait à standardiser la manière dont les portefeuilles HD génèrent et gèrent différents types de scripts de sortie. Supplanté par les output script descriptors, qui offrent une approche plus flexible et expressive pour décrire les scripts de sortie d'un portefeuille.

Termes associés : **BIP-0032** **SCRIPT**

BIP-0125

BIP

Définit le concept de *Replace-by-Fee* (RBF), permettant à l'émetteur de remplacer une transaction non confirmée par une autre version qui inclut des frais de transaction plus élevés. Le BIP-0125 offre un cadre pour le signalement de RBF dans une transaction et pour son acceptation par les nœuds du réseau.

Terme associé : **RBF - REPLACE-BY-FEE**

BIP-0126

BIP

Recommandations pour les transactions utilisant des inputs de types de scripts différents (P2PKH, P2SH, SegWit...). Le BIP-0126 vise à réduire les empreintes de portefeuille identifiables lors de la construction de transactions hétérogènes inévitables ou intentionnelles. Les transactions mélangeant des types de scripts révèlent des informations sur le portefeuille utilisé, ce qui nuit à la confidentialité. Ce document propose des bonnes pratiques pour minimiser ces fuites d'information lors de la construction de transactions hétérogènes.

Termes associés : **COINJOIN** **EMPREINTE DE PORTEFEUILLE**

BIP-0127

BIP

Définition d'un format standard de preuve de réserves basé sur des transactions Bitcoin intentionnellement invalides. Actuellement au statut de brouillon, le BIP-0127 permet aux custodians de prouver cryptographiquement qu'ils contrôlent certains UTXO sans avoir à les déplacer. La transaction de preuve inclut un premier input dit « d'engagement » (*commitment input*) dont le `txid` est remplacé par le hash SHA-256 d'un message de preuve. C'est cet input fictif qui rend la transaction invalide et impossible à confirmer sur la blockchain. Les autres inputs sont signés normalement et correspondent aux fonds détenus. La transaction dispose d'un unique output dont le montant est égal à la somme de tous les inputs, sans frais de minage. Le BIP-0127 s'appuie sur le format PSBT du BIP-0174 pour faciliter la construction des preuves avec l'infrastructure de portefeuilles existante, et définit également un format de fichier basé sur Protocol Buffers pour regrouper plusieurs preuves avec leurs métadonnées associées.

Termes associés : **PREUVE DE RÉSERVES** **UTXO**

BIP-0129

BIP

Proposition définissant le protocole BSMS (*Bitcoin Secure Multisig Setup*), un standard pour la mise en place sécurisée de portefeuilles multisignatures entre différents signataires matériels et logiciels. Le protocole utilise un schéma de chiffrement et une coordination par jeton (*token*) pour l'échange des clés publiques étendues et de la configuration du portefeuille. Il garantit que tous les cosignataires s'accordent sur les mêmes paramètres (seuil de signatures, descripteur, chemins de dérivation) avant de finaliser la création du portefeuille. Le BIP-0129 est adopté par plusieurs logiciels comme Nunchuk et Sparrow.

Termes associés : **MULTISIG** **PSBT**

BIP-0130

BIP

Proposition ajoutant le message P2P `sendheaders` au protocole réseau de Bitcoin. Ce message permet à un noeud de demander à son pair de lui annoncer les nouveaux blocs directement via des messages `headers`, plutôt que par des messages `inv` suivis d'une requête explicite. Ce mécanisme réduit la latence de propagation des blocs en éliminant un aller-retour réseau supplémentaire. Le noeud récepteur peut ainsi commencer à valider l'en-tête du bloc immédiatement après l'annonce. Le BIP-0130 est déployé et actif sur le réseau Bitcoin depuis 2016.

Termes associés : **COMPACT BLOCK RELAY** **RÉSEAU BITCOIN**

BIP-0131

BIP

Proposition de transactions dites de « coalescence » introduisant des inputs *wildcard* capables de consommer automatiquement tous les UTXO associés à une même adresse en une seule opération. Le BIP-0131 visait à simplifier la consolidation d'UTXO, mais a été rejeté en raison de problèmes de sécurité et d'incompatibilité avec le modèle de transaction existant de Bitcoin.

Termes associés : **CONSOLIDATION** **UTXO**

BIP-0132

BIP

Proposition de création d'un comité chargé d'accepter ou de rejeter les BIP selon un processus de vote structuré avec des membres désignés. Le BIP-0132 a été fermement rejeté par la communauté Bitcoin, car il était jugé incompatible avec la gouvernance décentralisée du projet. Ce BIP a été fermé.

Termes associés : **BIP** **BIP-0001**

BIP-0133

BIP

Proposition ajoutant le message P2P `feefilter` au protocole réseau de Bitcoin. Ce message permet à un nœud d'informer ses pairs du taux de frais minimum qu'il accepte pour le relais de transactions. Lorsqu'un nœud reçoit un `feefilter`, il s'abstient d'envoyer des transactions dont les frais sont inférieurs au seuil indiqué. Ce mécanisme réduit la consommation de bande passante en évitant la transmission de transactions qui seraient de toute façon rejetées par la mempool du pair. Le BIP-0133 est déployé et actif sur le réseau Bitcoin depuis la version 0.13.0 de Bitcoin Core.

Termes associés : **FRAIS DE TRANSACTION** | **MEMPOOL**

BIP-0134

BIP

Proposition de Tom Zander (Bitcoin Classic) introduisant un nouveau format de transaction flexible (« Flexible Transactions ») visant à résoudre le problème de malléabilité des transactions. Le BIP-0134 était un concurrent direct de SegWit, proposant une approche différente basée sur un encodage par tags. Il n'a jamais été adopté, le réseau ayant activé SegWit (BIP-0141) à la place.

Termes associés : **MALLÉABILITÉ - TRANSACTION** | **SEGWIT**

BIP-0135

BIP

Proposition généralisant le mécanisme de signalement par *version bits* défini dans le BIP-0009, avec davantage de flexibilité dans les paramètres d'activation des soft forks. Le BIP-0135 permettait de configurer les seuils d'activation, les périodes de signalement et les délais d'expiration de manière indépendante pour chaque proposition. Il n'a jamais été déployé sur Bitcoin.

Termes associés : **BIP-0009** | **SOFT FORK**

BIP-0136

BIP

Proposition d'un encodage Bech32 pour les références de position de transaction dans la blockchain, combinant la hauteur du bloc et l'index de la transaction. Le BIP-0136 offre un format compact et lisible pour identifier de manière unique une transaction confirmée, avec les avantages de Bech32 : détection d'erreurs intégrée et insensibilité à la casse. Le but était de remplacer l'usage des TXID pour les applications qui en avaient besoin. Ce BIP est resté en brouillon depuis 2017.

Termes associés : **BECH32 ET BECH32M** | **TXID - TRANSACTION IDENTIFIER**

BIP-0137

BIP

Propose un format standardisé pour signer des messages avec des clés privées Bitcoin et leurs adresses associées, afin de prouver la possession d'une adresse. Ce BIP vise à résoudre l'ambiguïté liée aux différents types d'adresses Bitcoin (P2PKH, P2SH-P2WPKH, P2WPKH...) lors de la signature d'un message. Il introduit une méthode permettant de distinguer explicitement ces formats d'adresses à travers les signatures. Ces signatures sont utiles pour diverses applications comme la preuve de fonds, l'audit, et d'autres utilisations nécessitant une authentification d'une adresse via sa clé privée. Le BIP-0322 a depuis introduit un nouveau format de signature qui permet d'étendre ce standard et de le généraliser pour n'importe quel type de script.

Termes associés : **BIP-0322** **P2PKH**

BIP-0140

BIP

Proposition de TXID normalisés excluant les données de signature du calcul de l'identifiant de transaction, afin de résoudre le problème de malléabilité. Le BIP-0140 constituait une approche alternative à SegWit pour obtenir des identifiants de transaction stables. Il a été fermé car SegWit (BIP-0141) a résolu le problème de malléabilité de manière différente, tout en apportant d'autres avantages.

Termes associés : **MALLÉABILITÉ - TRANSACTION** **TXID - TRANSACTION IDENTIFIER**

BIP-0141

BIP

Introduit le concept de témoin séparé (*Segregated Witness*) qui donnera son nom au soft fork SegWit. Le BIP-0141 introduit dans le protocole Bitcoin une modification majeure visant à résoudre le problème de malléabilité des transactions. SegWit sépare les témoins (données de signatures) du reste des données de transaction. Cette séparation est réalisée en insérant les témoins dans une structure de données distincte, engagée dans un nouvel arbre de Merkle, qui est lui-même référencé dans le bloc via la transaction coinbase, ce qui rend SegWit compatible avec les anciennes versions du protocole.

Terme associé : **SEGWIT**

BIP-0142

BIP

Proposition d'un format d'adresse pour SegWit basé sur l'encodage Base58Check existant. Le BIP-0142 a été supplanté par Bech32 (BIP-0173) qui offre une meilleure détection d'erreurs, une insensibilité à la casse et une compatibilité optimisée avec les QR codes. Ce format d'adresse n'a donc jamais été déployé sur le réseau Bitcoin.

Termes associés : **BECH32 ET BECH32M** **SEGWIT**

BIP-0143

BIP

Introduit une nouvelle manière de hacher la transaction pour la vérification des signatures dans les scripts post-SegWit. L'objectif est de minimiser les opérations redondantes lors de la vérification et d'inclure la valeur des UTXO en entrée dans la signature. Cela résout deux problèmes majeurs de l'algorithme de hachage de transaction original :

- La croissance quadratique du hachage des données avec le nombre de signatures ;

L'absence d'inclusion de la valeur de l'input dans la signature, ce qui posait un risque pour les hardware wallet*, notamment sur le fait de connaitre les frais engagés dans la transaction. Puisque la mise à jour SegWit, expliquée dans le BIP-0141, introduit une nouvelle forme de transactions dont le script ne sera pas vérifié par les vieux nœuds, le BIP-0143 en profite pour résoudre ce problème sans nécessiter de *hard fork*. Le BIP-0143 fait donc partie du *soft fork* SegWit.

Termes associés : **BIP-0141** **SEGWIT**

BIP-0144

BIP

Définit de nouveaux formats de messages réseaux et de sérialisations pour la diffusion des transactions et des blocs qui intègrent des structures de témoin séparés (SegWit). Le BIP-0144 établit notamment des mécanismes permettant aux pairs de signaler leur support pour SegWit et de relayer les structures de témoins sans compromettre la compatibilité avec les nœuds pas à jour.

Termes associés : **BIP-0141** **SEGWIT**

BIP-0145

BIP

Met à jour l'appel JSON-RPC `getblocktemplate` pour intégrer le support de SegWit, conformément au BIP-0141. Cette mise à jour permet aux mineurs de construire des blocs en tenant compte de la nouvelle mesure de « poids » des transactions et des blocs introduite par SegWit, et d'autres paramètres comme le calcul de la limite des sigops.

Termes associés : **BIP-0141** **SIGOPS**

BIP-0146

BIP

Proposition visant à corriger deux sources résiduelles de malléabilité des signatures dans les transactions Bitcoin, après l'adoption de l'encodage DER strict par le BIP-0066. La première règle, « LOW_S », impose que la valeur S des signatures ECDSA soit dans la moitié inférieure de l'ordre de la courbe, éliminant ainsi la malléabilité inhérente aux signatures ECDSA. La seconde règle, « NULLFAIL », exige que toute signature échouant à la vérification via `OP_CHECKSIG` ou `OP_CHECKMULTISIG` soit un tableau d'octets vide, empêchant ainsi le remplacement de signatures invalides par d'autres valeurs invalides.

Le BIP-0146 a le statut « fermé », car ces corrections ont été partiellement intégrées dans les règles de standardisation et dans SegWit (BIP-0141), qui a résolu le problème de malléabilité de manière plus complète.

Termes associés : **MALLÉABILITÉ - TRANSACTION** | **SEGWIT**

BIP-0147

BIP

Proposition incluse dans le soft fork SegWit visant à résoudre un vecteur de malléabilité lié à l'élément fictif (« *dummy element* ») consommé par les opérations `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY`. En raison d'un bug *off-by-one* historique (erreur de décalage unitaire), ces 2 opcodes suppriment un élément supplémentaire sur la pile en plus de leur fonction de base. Pour éviter une erreur, il est donc obligatoire d'inclure une valeur factice au début du `scriptSig` afin de satisfaire la suppression et outrepasser le bug. Cette valeur est inutile, mais elle doit forcément être là pour que le script soit valide. Le BIP-0011, qui a introduit le standard P2MS, conseillait de mettre un `OP_0` comme valeur inutile. Mais ce standard n'était pas imposé au niveau des règles de consensus, ce qui veut dire que n'importe quelle valeur pouvait y être placée, sans invalider la transaction. C'est en ça que `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY` contenaient un vecteur de malléabilité. Le BIP-0147 introduit une nouvelle règle de consensus, désignée sous le nom de `NULLDUMMY`, exigeant que cet élément fictif soit un tableau d'octets vide (`OP_0`). Toute autre valeur entraîne l'échec immédiat de l'évaluation du script. Cette modification s'applique aux scripts pré-SegWit ainsi qu'aux scripts P2WSH et nécessitait un soft fork.

Termes associés : **NULLDUMMY** | **OP_CHECKMULTISIG - 0XAE**

BIP-0148

BIP

Proposition introduite en mars 2017 par un développeur sous le pseudonyme de Shaolin Fry. L'objectif du BIP-0148 était de forcer l'activation de la mise à jour SegWit sur le protocole Bitcoin, face à la stagnation de la signalisation de ce soft fork par les mineurs via la méthode du BIP-0009. Le BIP-0148 suggérait la mise en œuvre d'un UASF (*User-Activated Soft Fork*) pour activer SegWit de force par les nœuds le 15 novembre 2017, si les mineurs n'avaient pas verrouillé SegWit d'ici au 1er août 2017. Si l'adoption de l'UASF du BIP-0148 avait eu lieu, les nœuds du réseau Bitcoin auraient refusé les blocs ne signalant pas le support à SegWit, exerçant ainsi une pression sur les mineurs pour qu'ils adoptent la mise à jour. Bien que ce BIP historique n'ait finalement pas été activé, il a joué un rôle déterminant dans la réussite de l'adoption de SegWit, en contraignant les mineurs à verrouiller le soft fork via le BIP-0091. À plus long terme, le BIP-0148 a établi un précédent important, démontrant l'influence que peuvent exercer les utilisateurs via leurs nœuds complets sur les décisions de gouvernance du protocole Bitcoin.

Terme associé : **UASF**

BIP-0149

BIP

Proposition de Shaolin Fry pour un nouveau déploiement de SegWit (BIP-0141, BIP-0143 et BIP-0147) en utilisant la méthode d'activation du BIP-0008 avec `LOT=true`, si le déploiement initial de SegWit via le BIP-0009 échouait à s'activer avant le 15 novembre 2017. Contrairement à la méthode du BIP-0009, où un échec de signalisation entraîne l'abandon de l'activation, le BIP-0149 visait à activer SegWit le 4 juillet 2018, que les mineurs aient atteint le seuil de signalisation de 95 % ou non. Pendant la période de huit mois entre novembre et juillet, les nœuds auraient eu la possibilité d'implémenter le BIP-0149, afin d'assurer une activation de SegWit par la majorité économique du réseau si l'activation par les mineurs ne se produisait pas (UASF). Une fois le premier ajustement de difficulté atteint après le 4 juillet 2018, l'activation serait passée en `LOCKED_IN`, et SegWit aurait été activé au cycle d'ajustement suivant. Contrairement au BIP-0148, qui prévoyait une activation de SegWit imposée par les utilisateurs ou une majorité de mineurs, le BIP-0149 suggérait une méthode d'activation plus progressive et mesurée, bien qu'elle demeurât résolument offensive, selon les principes du BIP-0008. Alors que le BIP-0148 laissait présager un conflit avec une séparation de la blockchain, le BIP-0149 écartait cette éventualité, en acceptant les blocs ne signalant pas SegWit, sauf action délibérée d'un mineur (sans incitation). Le BIP-0149 était donc un mécanisme d'activation de SegWit moins conflictuel que le BIP-0148, favorisant une adoption plus progressive et moins risquée pour le système. Ni le BIP-0148 ni le BIP-0149 n'ont finalement été mis en œuvre, SegWit ayant été activé grâce à un MASF, notamment sous l'impulsion du BIP-0091.

Termes associés : **BIP-0009** | **BIP-0091** | **BIP-0148** | **MÉTHODE D'ACTIVATION** | **UASF**

BIP-0150

BIP

Propose un mécanisme d'authentification entre les pairs sur le réseau Bitcoin pour renforcer la sécurité et garantir la propriété des nœuds. Il permet aux opérateurs de nœuds de restreindre l'accès à certains services ou d'accorder des priorités de flux de données uniquement à des pairs spécifiques, en s'authentifiant mutuellement pour éviter les attaques de type MITM. Ce BIP a été fermé (« *Closed* »), mais il aura servi d'enseignement pour le BIP-0324 (*P2P transport V2*) qui est aujourd'hui implémenté en option dans Bitcoin Core.

Terme associé : **P2P TRANSPORT V2**

BIP-0151

BIP

Propose un protocole pour chiffrer les communications P2P entre pairs sur le réseau Bitcoin, afin de renforcer la sécurité et la confidentialité. Son objectif est notamment de prévenir les manipulations du trafic et les attaques de surveillance de masse. Finalement, le BIP-0151 a été remplacé par le BIP-0324 (*P2P transport V2*) qui est aujourd'hui implémenté en option dans Bitcoin Core.

Terme associé : **P2P TRANSPORT V2**

BIP-0152

BIP

Proposition « *Compact Block Relay* » visant à réduire la bande passante nécessaire pour la transmission des blocs sur le réseau Bitcoin. Adopté en novembre 2016 dans la version 0.13.0 de Bitcoin Core, ce protocole permet de communiquer les informations des blocs de manière compacte, en se basant sur l'hypothèse que les nœuds ont déjà une grande partie des transactions d'un bloc récent dans leur mempool. Plutôt que de transmettre chaque transaction intégralement, ce qui constituerait un doublon, le BIP-0152 propose d'envoyer uniquement de courts identifiants pour les transactions déjà connues des pairs, accompagnés de quelques transactions sélectionnées (notamment la transaction coinbase et celles que le nœud est susceptible de ne pas connaître). Le nœud peut ensuite demander à ses pairs les éventuelles transactions manquantes. Compact Block Relay permet ainsi de diminuer la quantité de données échangées lors de la propagation des blocs, ce qui réduit ainsi les pics de bande passante et améliore l'efficacité globale du réseau.

Termes associés : **COINBASE** **MEMPOOL**

BIP-0154

BIP

Proposition d'un mécanisme de limitation de débit via des défis cryptographiques imposés par les pairs sur le réseau P2P de Bitcoin. Le BIP-0154 visait à atténuer les attaques par déni de service en exigeant des nœuds demandeurs qu'ils résolvent un petit défi de preuve de travail avant d'obtenir une réponse. Cette approche n'a jamais été déployée.

Termes associés : **DOS - DENIAL OF SERVICE** **RÉSEAU BITCOIN**

BIP-0155

BIP

Proposition d'amélioration de Bitcoin introduisant un nouveau format pour les messages qui transmettent des adresses de nœuds qui acceptent des connexions entrantes. Cette proposition permet de supporter des adresses de plus grande taille, notamment pour faciliter l'intégration de protocoles réseau comme Tor V3 ou I2P, dont les adresses sont plus longues que celles supportées par l'ancien message `addr`. Cette amélioration est également connue sous le nom de `addrv2`.

Termes associés : **ADDRV2** | **I2P**

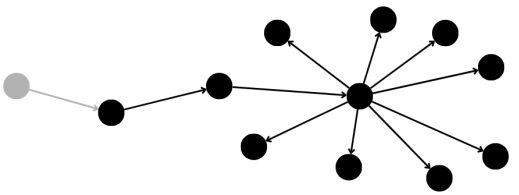
BIP-0156

BIP

Proposition, connue sous le nom de Dandelion, qui vise à améliorer la confidentialité du routage des transactions dans le réseau Bitcoin pour contrer la désanonymisation. Dans le fonctionnement classique de Bitcoin, les transactions sont immédiatement diffusées à de multiples nœuds. Si un observateur est en capacité de voir chaque transaction relayée par chaque nœud sur le réseau, il peut supposer que le premier nœud à envoyer une transaction est également le nœud d'origine de cette transaction, et donc que celle-ci provient de l'opérateur du nœud. Ce phénomène peut potentiellement permettre à des observateurs de lier des transactions, normalement anonymes, avec des adresses IP.

L'objectif du BIP-0156 est de traiter ce problème. Pour ce faire, il introduit une phase supplémentaire dans la diffusion permettant de préserver l'anonymat avant la propagation publique. Ainsi, Dandelion utilise d'abord une phase de « tige » où la transaction est envoyée à travers un chemin aléatoire de nœuds, avant d'être diffusée à l'ensemble du réseau dans la phase de « capitule ». La tige et le capitule sont des références au comportement de la propagation de la transaction à travers le réseau, qui ressemble à la forme d'un pissenlit (« *a dandelion* » en anglais).

Cette méthode de routage brouille la piste menant au nœud source, rendant difficile de retracer une transaction via le réseau jusqu'à son origine. Dandelion améliore donc la confidentialité en limitant la capacité des adversaires à désanonymiser le réseau. Cette méthode est d'autant plus efficace lorsque la transaction croise durant la phase de « tige » un nœud qui chiffre ses communications réseau, comme avec Tor ou *P2P Transport V2*. Le BIP-0156 a été rejeté et n'a pas été intégré à Bitcoin Core.



Terme associé : **P2P TRANSPORT V2**

BIP-0157

BIP

Proposition qui définit le protocole pair-à-pair permettant aux clients légers de télécharger des *compact block filters* depuis les nœuds complets. Ce BIP introduit six nouveaux messages P2P : `getcfilters` et `cfilter` pour demander et recevoir les filtres, `getcfheaders` et `cfheaders` pour les en-têtes de filtres, ainsi que `getcfcheckpt` et `cfcheckpt` pour des points de contrôle espacés tous les 1 000 blocs.

Le BIP-0157 s'appuie sur un système d'en-têtes de filtres chaînés, similaire aux en-têtes de blocs. Chaque en-tête est le double SHA-256 de la concaténation du hash du filtre avec l'en-tête précédent, ce qui permet au client de vérifier l'authenticité de la chaîne de filtres. En cas de conflit entre pairs, le client peut télécharger le bloc complet, recalculer le filtre correct et identifier le pair fautif.

Ce BIP requiert le BIP-0158 qui spécifie la construction des filtres eux-mêmes. Rédigé par Olaoluwa Osuntokun, Alex Akselrod et Jim Posen, il est déployé dans Bitcoin Core.

Termes associés : **BIP-0158** **COMPACT BLOCK FILTERS**

BIP-0158

BIP

Proposition qui définit la structure et la construction des *compact block filters* utilisés par le protocole BIP-0157. Le format retenu repose sur les *Golomb-Coded Sets* (GCS), une structure probabiliste plus compacte que les filtres de Bloom du BIP-0037.

Le filtre de type `basic` (`0×00`) contient, pour chaque transaction d'un bloc, les scripts de sortie dépensés en input (sauf la coinbase) et les `scriptPubKey` de chaque sortie (sauf les `OP_RETURN`). Les éléments sont hachés avec SipHash (paramètres `c = 2`, `d = 4`), la clé étant les 16 premiers octets du hash du bloc. Les valeurs hachées sont projetées dans l'intervalle $[0, N \times M)$, triées, puis les différences successives sont compressées par codage de Golomb-Rice.

Les paramètres retenus sont $P = 19$ et $M = 784\,931$, optimisés pour minimiser la bande passante globale en tenant compte à la fois de la taille des filtres et du nombre de blocs téléchargés à cause des faux positifs. Ce BIP introduit également le bit de service `NODE_COMPACT_FILTERS` (`1 < 6`). Il a été rédigé par Olaoluwa Osuntokun et Alex Akselrod.

Termes associés : **BIP-0157** **GCS - GOLOMB-CODED SET**

BIP-0159

BIP

Proposition définissant le bit de service `NODE_NETWORK_LIMITED` (bit 10) pour les nœuds élagués du réseau Bitcoin. Ce bit indique qu'un nœud est capable de servir au minimum les 288 derniers blocs de la blockchain (environ 2 jours), même sans disposer de l'historique complet. Avant ce BIP, les nœuds élagués ne pouvaient pas signaler leur capacité partielle de service, ce qui les excluait du relais de blocs. Le BIP-0159 permet ainsi aux nœuds élagués de participer activement à la propagation des blocs récents, améliorant la décentralisation du réseau sans nécessiter le stockage intégral de la chaîne.

Termes associés : **NOEUD ÉLAGUÉ** | **RÉSEAU BITCOIN**

BIP-0171

BIP

Définition d'un format d'API standard pour les données de taux de change Bitcoin, permettant aux applications d'interroger différents fournisseurs de prix de manière uniforme et interopérable. Le BIP-0171 spécifiait les *endpoints* REST, les formats de réponse JSON et les paires de devises supportées, visant à standardiser l'accès aux données de marché. Cette proposition a été fermée, les fournisseurs de données ayant développé leurs propres API sans adopter de standard commun.

Termes associés : **API** | **BIP**

BIP-0172

BIP

Proposition de définir officiellement le satoshi (sat) comme la sous-unité standard de Bitcoin et de promouvoir son utilisation dans les interfaces utilisateur. Le BIP-0172 vise à clarifier la dénomination en établissant que 1 bitcoin = 100 000 000 satoshis, et recommande l'affichage des montants en satoshis pour les petites valeurs, afin d'éviter la confusion liée aux nombres décimaux.

Termes associés : **MILLIBITCOIN** | **SAT - SATOSHI**

BIP-0173

BIP

Introduit le format d'adresse bech32 pour les adresses SegWit V0. Ce format d'adresse est caractérisé par le préfixe `bc1q`. Le format bech32 offre plusieurs avantages :

- Il demande moins d'espace dans les codes QR ;
- Il est plus facilement interprétable par les humains ;
- Il dispose d'un mécanisme innovant pour la somme de contrôle qui est plus performant et permet de détecter et potentiellement de modifier automatiquement les fautes de frappe.

Ces caractéristiques facilitent l'utilisation des adresses de réception tout en minimisant les risques d'erreurs.

Terme associé : **BECH32 ET BECH32M**

BIP-0174

BIP

Définit le format PSBT (*Partially Signed Bitcoin Transaction*), un standard pour échanger des transactions Bitcoin non finalisées entre différents acteurs tels que des portefeuilles logiciels, des hardware wallets ou des coordinateurs multisig. Le PSBT permet de séparer les étapes de création, de mise à jour, de signature et de finalisation d'une transaction. Le protocole définit six rôles distincts : le *Creator* crée la transaction de base, l'*Updater* ajoute les informations nécessaires (UTXO, scripts), le *Signer* appose sa signature, le *Combiner* fusionne les signatures partielles, le *Finalizer* construit le script de déverrouillage final, et l'*Extractor* produit la transaction prête à être diffusée.

Ce format est essentiel pour les multisigs, les hardware wallets, les coinjoins, et tout autre workflow où la signature ne se fait pas sur le même appareil que la construction de la transaction.

Termes associés : **BIP-0370** **PSBT**

BIP-0175

BIP

Description du protocole Pay-to-Contract (P2C) permettant d'engager un contrat dans une clé publique via un tweak cryptographique. Le payeur modifie la clé publique du destinataire en y incorporant le hash d'un contrat, créant une nouvelle clé qui ne peut être dépensée que par le destinataire. Ce mécanisme permet de prouver qu'un paiement est lié à un contrat spécifique sans révéler le contrat sur la blockchain.

Termes associés : **BIP** **TWEAK**

BIP-0176

BIP

Proposition d'utiliser le « bit » (100 satoshis, soit un millionième de bitcoin) comme dénomination standard dans les interfaces utilisateur. Le BIP-0176 argumente que le bit offre un bon compromis entre précision et lisibilité pour les montants courants. Cette proposition est en concurrence avec le mouvement vers le satoshi comme unité de référence principale, ce dernier gagnant davantage de traction dans l'écosystème.

Termes associés : **MILLIBITCOIN** **SAT - SATOSHI**

BIP-0177

BIP

Proposition de redéfinir l'unité de base de Bitcoin pour que « 1 bitcoin » corresponde à l'actuel satoshi, éliminant totalement les décimales. Sous ce nouveau système, la valeur actuelle de 1 BTC deviendrait 100 000 000 de la nouvelle unité de base. Le BIP-0177 vise à simplifier l'expérience utilisateur en supprimant la confusion liée aux fractions décimales, particulièrement problématique pour les nouveaux utilisateurs.

Cette modification a été adoptée par certaines applications, notamment Bitkit, le portefeuille développé par Synonym dont le CEO, John Carvalho, est également l'auteur du BIP-0177. Dans les faits, cependant, la grande majorité des logiciels de portefeuille conservent pour l'instant le standard 1 bitcoin = 100 000 000 satoshis.

Termes associés : **BIP** **SAT - SATOSHI**

BIP-0178

BIP

Proposition d'une extension du format WIF (Wallet Import Format) avec un octet de version supplémentaire indiquant le type de script associé à la clé privée. Le BIP-0178 permettrait de distinguer automatiquement si une clé est destinée à être utilisée avec P2PKH, P2SH-P2WPKH, P2WPKH ou un autre type de script, évitant ainsi les erreurs lors de l'importation de clés entre portefeuilles.

Termes associés : **SCRIPT** **WIF - WALLET IMPORT FORMAT**

BIP-0179

BIP

Document informatif proposant de remplacer le terme « adresse » par « invoice » (ou « *Bitcoin Invoice Address* ») pour désigner les identifiants de réception de paiement dans Bitcoin. Le BIP-0179 part du constat que le mot « adresse » suggère une information permanente (comme une adresse postale ou e-mail), alors que les adresses Bitcoin sont conçues pour un usage unique. Le terme « invoice » est proposé pour mieux refléter ce caractère éphémère et lié à une transaction spécifique. Les variantes autorisées sont « *Bitcoin Invoice Address* », « *Bitcoin Invoice* », « *Invoice Address* » et « *Invoice* », le suffixe « *Address* » étant toléré comme mesure transitoire.

Termes associés : **BIP** **BIP-0021**

BIP-0180

BIP

Proposition d'un format de preuve de fraude pour les blocs dépassant la taille ou le poids maximum autorisé. Le BIP-0180 permettrait aux nœuds légers de vérifier qu'un bloc respecte les limites de taille et de poids sans avoir à le télécharger entièrement, en recevant une preuve compacte de violation. Cette approche visait à renforcer la sécurité des clients légers. La proposition a été fermée.

Termes associés : **MAX_BLOC_SIZE** **NOEUD COMPLET**

BIP-0197

BIP

Proposition des HTLCC (*Hashed Time-Locked Collateral Contracts*), une extension des HTLC (*Hashed Time-Locked Contracts*) ajoutant un mécanisme de collatéral. Le BIP-0197 est conçu pour créer des accords de dette entre deux parties (un emprunteur et un prêteur), où du collatéral en bitcoins est verrouillé dans un P2SH pour garantir un prêt dont le principal est libellé sur une autre blockchain. En cas de non-remboursement, le collatéral peut être liquidé via un atomic swap ou saisi selon des conditions prédéfinies.

Termes associés : **ATOMIC SWAP** | **HTLC**

BIP-0199

BIP

Formalisation de la construction des HTLC (*Hashed Time-Locked Contracts*) dans les transactions Bitcoin on-chain. Le BIP-0199 décrit le mécanisme standard utilisé pour les *atomic swaps* et les canaux de paiement Lightning : un *hashlock* combiné à un *timelock* permettant le remboursement conditionnel. Le destinataire doit révéler un secret (préimage du hash) avant l'expiration du délai pour réclamer les fonds, sinon l'expéditeur peut les récupérer.

Termes associés : **ATOMIC SWAP** | **HTLC**

BIP-0300

BIP

Proposition de Paul Sztorc et CryptAxe définissant le mécanisme de « *Hashrate Escrows* » (séquestre par le hashrate), qui constitue la première brique technique des drivechains.

Ce BIP permet de créer des sidechains dont l'ancrage bilatéral avec Bitcoin est géré directement par les mineurs, sans recourir à une fédération de signataires. Les bitcoins envoyés vers une sidechain sont verrouillés dans une sortie spéciale utilisant `OP_DRIVECHAIN` (redéfinition de `OP_NOP5`). Pour retirer des fonds vers la chaîne principale, un « *bundle* de retrait » est proposé puis soumis à un vote des mineurs s'étalant sur 26 300 blocs (environ 6 mois). Chaque mineur peut, bloc par bloc, voter en faveur d'un bundle, s'abstenir, ou déclencher une alarme qui réduit le score de tous les bundles. Le retrait n'est exécuté que si le bundle atteint 13 150 votes favorables, soit environ 50 % du hashrate sur la période. Cette lenteur est volontaire : elle rend toute tentative de vol publiquement visible pendant des mois, laissant le temps à la communauté de réagir.

Le BIP-0300 définit six messages de consensus (M1 à M6) couvrant la proposition et l'activation de nouvelles sidechains, les dépôts et les retraits.

Ce BIP a fait l'objet de débats importants : ses partisans y voient un moyen d'ajouter des fonctionnalités (confidentialité, smart contracts, scalabilité) sur des sidechains sans modifier la couche de base ; ses détracteurs soulèvent des questions sur le modèle de sécurité, puisqu'une attaque à 51 % maintenue suffisamment longtemps pourrait permettre le vol de l'intégralité des fonds d'une sidechain, et sur un risque de centralisation du minage. Le BIP-0300 est

donc toujours en brouillon.

Termes associés : **DRIVECHAIN** **SIDECHAIN**

BIP-0301

BIP

Proposition de Paul Sztorc et CryptAxe définissant le « *Blind Merged Mining* » (BMM), un mécanisme de minage fusionné aveugle pour les drivechains. Le BMM permet aux mineurs de Bitcoin de percevoir les frais de transaction des sidechains en BTC sur Bitcoin, sans avoir besoin d'exécuter le logiciel de chaque sidechain. Le BIP-0301 complète le BIP-0300 pour former la base technique complète des drivechains.

Termes associés : **BIP-0300** **MINAGE FUSIONNÉ**

BIP-0310

BIP

Proposition d'extensions au protocole Stratum utilisé pour la communication entre les mineurs individuels (les hacheurs) et les pools de minage. Le BIP-0310 introduit un mécanisme générique de négociation de fonctionnalités via un nouveau message `mining.configure`, qui permet au mineur et au serveur de s'accorder sur les extensions supportées. La principale extension spécifiée est le *version rolling*, qui autorise les mineurs à modifier certains bits du champ `nVersion` de l'en-tête de bloc. Le BIP définit également d'autres extensions comme la difficulté minimale et le `subscribe-extranonce`. Il constitue un prédécesseur conceptuel de Stratum V2, qui a repris et approfondi ces améliorations avec l'ajout du chiffrement des communications et de la sélection des transactions par les hacheurs.

Termes associés : **STRATUM** **STRATUM V2**

BIP-0320

BIP

Proposition de réserver certains bits du champ `nVersion` des en-têtes de blocs pour un usage général, distinct du mécanisme de signalement de soft forks défini par le BIP-0009. La motivation principale du BIP-0320 est de permettre l'utilisation de l'Overt ASICBoost de manière propre : cette technique d'optimisation du minage nécessite de modifier les bits de version du bloc, ce qui peut interférer avec le signalement des soft forks. En réservant des bits dédiés, les mineurs peuvent utiliser ASICBoost sans ambiguïté avec les mécanismes d'activation.

Termes associés : **BIP-0009** **OVERT ASICBOOST**

BIP-0321

BIP

Proposition mettant à jour le schéma d'URI `bitcoin:` défini par le BIP-0021 afin de prendre en charge les adresses SegWit au format bech32 et bech32m. Le BIP-0021 original avait été conçu pour les adresses Base58, qui sont sensibles à la casse. Les adresses bech32 étant insensibles à la casse, le BIP-0321 adapte la spécification pour permettre leur utilisation correcte dans les URI, notamment dans les QR codes où l'encodage en majuscules est plus efficace. Ce BIP garantit la compatibilité du schéma d'URI Bitcoin avec les formats d'adresses modernes.

Termes associés : **BIP-0021** **SEGWIT**

BIP-0322

BIP

Propose un nouveau standard en remplacement du BIP-0137 pour la signature de messages avec des clés privées Bitcoin et leurs adresses associées, afin de prouver la possession d'une adresse. Ces signatures sont utiles pour diverses applications comme la preuve de fonds, l'audit, et d'autres utilisations nécessitant une authentification d'une adresse via sa clé privée. Par rapport au BIP-0137, le BIP-0322 étend le standard de signature de messages au-delà des adresses classiques, en utilisant une approche établie sur les scripts. Il permet aux logiciels de portefeuille de signer un message pour n'importe quel script qu'ils pourraient débloquer pour dépenser des bitcoins. Pour ce faire, la méthode consiste à signer un texte en produisant une signature pour une transaction Bitcoin virtuelle. Pour les adresses P2PKH traditionnelles, le BIP-0322 reste compatible avec le format de signature traditionnel.

Termes associés : **BIP-0137** **P2PKH**

BIP-0324

BIP

Introduit une nouvelle version du protocole de transport Bitcoin P2P intégrant le chiffrement opportuniste pour améliorer la confidentialité et la sécurité des communications entre les nœuds. Le transport P2P V2 du BIP-0324 a été inclus en option (désactivé par défaut) dans la version 26.0 de Bitcoin Core, déployée en décembre 2023. Il peut être activé avec l'option `v2transport=1` dans le fichier de configuration. Cette amélioration remplace le BIP-0151.

Terme associé : **P2P TRANSPORT V2**

BIP-0325

BIP

Proposition définissant les Signets, des réseaux de test pour Bitcoin dans lesquels la production de blocs est contrôlée par un ensemble de signataires autorisés, en plus de la preuve de travail. Contrairement au testnet classique, où n'importe qui peut miner et provoquer des réorganisations imprévisibles, les blocs sur Signet doivent contenir une signature valide émise par des clés désignées. Cela rend le réseau stable, prévisible et adapté au développement et aux tests d'applications. Le signet par défaut est maintenu par Kalle Alm et AJ Towns. Il est possible de créer des signets personnalisés avec ses propres clés de signature.

Termes associés : **SIGNET** **TESTNET**

BIP-0326

BIP

Proposition d'amélioration destinée aux développeurs de logiciels de portefeuille Bitcoin prenant en charge les transactions Taproot. Son but principal est de renforcer la confidentialité des protocoles de seconde couche qui pourraient utiliser des PTLC (*Point Time Locked Contracts*), comme les CoinSwap, le Lightning Network ou les DLC (*Discreet Log Contracts*). Pour ce faire, cette proposition vise à créer du déni plausible en configurant automatiquement le champ `nSequence` des transactions Taproot de la même manière que le champ `nLocktime` était configuré dans les autres types de transactions afin de décourager les attaques de *fee sniping*. Autrement dit, le BIP-0326 demande aux logiciels de portefeuille d'utiliser le champ `nSequence` plutôt que le champ `nLocktime` pour prévenir les attaques de *fee sniping*, afin d'offrir une confidentialité accrue pour tous les protocoles *off-chain* utilisant ce champ de manière similaire. Ainsi, une transaction Taproot avec une valeur spécifique dans le champ `nSequence` pourrait être soit une dépense somme toute classique d'un portefeuille, soit une transaction de règlement d'un protocole de seconde couche avec un verrouillage temporel, rendant ces deux cas indiscernables. Si cette proposition d'amélioration est appliquée massivement par les développeurs de logiciels de portefeuille Bitcoin, cela améliorerait grandement la confidentialité et la fongibilité de Bitcoin au global.

Terme associé : **FEE SNIPING**

BIP-0327

BIP

Définit MuSig2, un schéma de multi-signature dans lequel *n* participants produisent conjointement une unique signature de Schnorr, indiscernable *on-chain* d'une signature individuelle classique. Le protocole fonctionne en deux tours : un échange de *nonces* publics, puis la génération de signatures partielles qui sont combinées en une signature finale. L'agrégation de clés est non interactive : la clé publique agrégée peut être calculée par n'importe qui à partir des clés publiques individuelles. MuSig2 permet ainsi un *multisig n-of-n* qui apparaît comme une dépense à clé unique sur la blockchain, ce qui améliore la confidentialité et réduit les frais de transaction.

Terme associé : **BIP-0340**

BIP-0328

BIP

Définit comment construire une clé publique étendue (xpub) BIP-0032 synthétique à partir d'une clé publique agrégée MuSig2 (BIP-0327), et comment l'utiliser pour la dérivation de clés enfants. Plutôt que d'exiger que chaque participant dérive ses clés individuellement avant de les agréger à chaque niveau, le BIP-0328 permet de dériver directement des clés enfants non endurcies à partir d'une seule xpub synthétique construite sur la clé agrégée. Cette xpub utilise un code de chaîne fixe et standardisé. Ce mécanisme simplifie considérablement la génération d'adresses de réception dans les portefeuilles multisig MuSig2, en réduisant le stockage et le calcul nécessaires. Lors de la signature, les participants doivent calculer les tweaks de dérivation BIP-0032 correspondants et les fournir dans le contexte de session MuSig2.

Termes associés : **BIP-0032** **BIP-0327**

BIP-0329

BIP

Proposition définissant un format standard en JSON Lines pour l'exportation et l'importation des étiquettes de portefeuille. Ce format couvre les étiquettes de transactions, d'adresses, de clés publiques étendues et d'UTXO. L'objectif est de permettre aux utilisateurs de migrer leurs données d'étiquetage entre différents logiciels de portefeuille sans perte d'information. Le BIP-0329 répond à un besoin pratique, car l'étiquetage des transactions et des adresses est essentiel pour la gestion comptable et la confidentialité des utilisateurs.

Terme associé : **ÉTIQUETAGE**

BIP-0330

BIP

Proposition connue sous le nom de « *Erlay* » qui a pour objectif d'optimiser la propagation des transactions non confirmées dans le réseau Bitcoin. Actuellement, les transactions sont diffusées à tous les pairs d'un nœud, ce qui entraîne des redondances et une surutilisation de la bande passante. Le BIP-0330 propose de limiter cette diffusion à 8 pairs par défaut, puis d'utiliser un mécanisme de réconciliation pour partager efficacement les transactions manquantes. Erlay réduit la consommation de bande passante d'environ 40 %. Il permet également de ne pas avoir une augmentation linéaire de la consommation de bande passante avec le nombre de pairs connectés au nœud.

Terme associé : **ERLAY**

BIP-0331

BIP

Proposition (brouillon) décrivant un protocole P2P pour le relais de paquets de transactions apparentées (*ancestor packages*). Ce BIP permet de relayer une transaction parente à faibles frais conjointement avec sa transaction enfant à frais plus élevés, rendant ainsi le mécanisme CPFP (*Child Pays For Parent*) efficace dès le niveau du relais réseau. Sans ce protocole, une transaction parente dont les frais sont inférieurs au minimum du mempool ne peut pas être propagée, même si une transaction enfant est prête à compenser. Le BIP-0331 résout ce problème en permettant aux nœuds d'évaluer le taux de frais de l'ensemble du paquet plutôt que de chaque transaction individuellement.

Termes associés : **CPFP - CHILD PAY FOR PARENT** **MEMPOOL**

BIP-0337

BIP

Proposition d'un format de compression pour les transactions Bitcoin, réduisant significativement leur taille en omettant les données redondantes ou déductibles du contexte.

Termes associés : **BIP** **TXID - TRANSACTION IDENTIFIER**

BIP-0338

BIP

Proposition d'un message P2P `disabletx` permettant à un noeud d'indiquer à ses pairs qu'il ne souhaite pas recevoir de transactions relayées sur cette connexion. Le BIP-0338 visait les connexions dédiées uniquement au relais de blocs, réduisant ainsi la bande passante et la surface d'attaque. Cette proposition a été fermée, la fonctionnalité ayant été intégrée autrement dans le protocole réseau de Bitcoin Core.

Termes associés : **BIP** **RÉSEAU BITCOIN**

BIP-0339

BIP

Proposition modifiant le relais de transactions pour utiliser le `wtxid` (*witness transaction identifier*) au lieu du `txid` classique lors de l'annonce et de la requête de transactions entre pairs. Avant ce BIP, un pair malveillant pouvait envoyer des témoins (*witnesses*) invalides pour un `txid` valide, gaspillant ainsi la bande passante du noeud récepteur. En utilisant le `wtxid`, qui intègre les données de témoin SegWit dans son calcul, chaque variante de témoin possède un identifiant unique, ce qui empêche cette classe d'attaques par déni de service. Les noeuds signalent leur support via le message `wtxidrelay` échangé lors de la poignée de main de version.

Termes associés : **SEGWIT** **WTXID**

BIP-0340

BIP

Définit le schéma de signatures de Schnorr pour la courbe secp256k1 de Bitcoin, activé avec SegWit v1 (Taproot) en novembre 2021. Par rapport à ECDSA, les signatures de Schnorr offrent plusieurs avantages : elles sont plus simples mathématiquement, disposent d'une preuve formelle de sécurité, et permettent nativement l'agrégation de clés (MuSig2), la vérification par lots et des multisignatures plus compactes. Le BIP-0340 utilise des signatures de 64 octets (sans encodage DER) et des clés publiques *x-only* de 32 octets, ce qui réduit la taille des données on-chain.

Termes associés : **BIP-0341** **LIBSECP256K1**

BIP-0341

BIP

Définit les règles de dépense de SegWit version 1, communément appelé « Taproot ». Ce BIP introduit deux chemins de dépense distincts : le *key-path*, qui permet de dépenser avec une simple signature de Schnorr, et le *script-path*, qui permet de révéler et d'exécuter un script depuis un arbre de Merkle (MAST). Les sorties Taproot utilisent des clés publiques « tweakées » qui combinent une clé interne avec la racine de l'arbre MAST.

L'avantage de cette structure est que toutes les sorties Taproot ont une apparence identique on-chain (une simple clé publique de 32 octets), ce qui améliore considérablement la confidentialité des utilisateurs. Le BIP-0341 a été activé au bloc 709 632 en novembre 2021 via la méthode Speedy Trial.

Termes associés : **BIP-0340** **TAPROOT**

BIP-0342

BIP

Définit Tapscript, le langage de script utilisé pour les dépenses par chemin de script (*script-path*) dans Taproot (BIP-0341). Tapscript apporte plusieurs modifications par rapport au script Bitcoin traditionnel. Il remplace `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY` par `OP_CHECKSIGADD`, un nouvel opcode qui facilite la vérification par lots des signatures de Schnorr. Le BIP-0342 introduit également les opcodes `OP_SUCCESS`, qui réservent des plages d'opcodes pour de futures extensions du protocole sans nécessiter de soft fork complexe. La validation des signatures repose sur le schéma de Schnorr défini dans le BIP-0340. De plus, Tapscript supprime la limite de 10 000 octets par script et la limite de 201 opcodes non-push qui s'appliquaient aux scripts Legacy, offrant ainsi plus de flexibilité aux développeurs.

Termes associés : **BIP-0341** **TAPSCRIPT**

BIP-0343

BIP

Spécifie un mécanisme d'activation alternatif pour le soft fork Taproot, basé sur le BIP-0008 avec le paramètre `LOT=true` (activation obligatoire). Ce BIP, proposé par Shinobius et Michael Folkson, a été conçu en parallèle du « Speedy Trial » déployé dans Bitcoin Core, pour garantir que les mineurs ne puissent pas bloquer indéfiniment l'activation d'un soft fork disposant d'un consensus écrasant de la communauté. La période de signalisation commençait au bloc 681 408 avec un seuil de 90 %, et prévoyait un signalement obligatoire avant le bloc 760 032 si le verrouillage n'était pas atteint. En pratique, Taproot a été verrouillé via le Speedy Trial de Bitcoin Core au bloc 687 285 en juin 2021 et activé en novembre, rendant le BIP-0343 caduc avant que sa phase de signalement obligatoire ne soit nécessaire.

Termes associés : **BIP-0341** **SPEEDY TRIAL**

BIP-0345

BIP

Proposition d'ajout de deux opcodes, `OP_VAULT` et `OP_VAULT_RECOVER`, permettant de créer des « *vaults* » (coffres-forts) natifs sur Bitcoin. Un vault est un mécanisme de sécurité qui impose un délai d'attente lors du retrait de fonds : pendant ce délai, le propriétaire peut annuler la transaction et rediriger les fonds vers une adresse de récupération prédéfinie. Ce système protège contre le vol en offrant une fenêtre de réaction même si les clés principales sont compromises. Proposé par James O'Beirne, le BIP-0345 a été fermé au profit de `OP_CHECKCONTRACTVERIFY` (BIP-0443).

Termes associés : **COVENANT** **OPCODES**

BIP-0346

BIP

Propose l'ajout de l'opcode `OP_TXHASH` pour Tapscript, qui place sur la pile le hash de certains champs sélectionnés de la transaction courante. Le choix des champs à inclure dans le hash est déterminé par un paramètre de type *TxFieldSelector* qui permet de spécifier précisément quelles parties de la transaction (entrées, sorties, montants, séquences...) doivent être couvertes. Combiné avec `OP_CHECKSIGFROMSTACK` (BIP-0348), `OP_TXHASH` permettrait une forme flexible d'introspection de transaction et de covenants, sans nécessiter un opcode dédié pour chaque cas d'usage. Proposé par Steven Roose.

Termes associés : **COVENANT** **OPCODES**

BIP-0347

BIP

Proposition pour réactiver l'opcode `OP_CAT` dans Tapscript uniquement. `OP_CAT` concatène les deux éléments au sommet de la pile en un seul (c'est-à-dire qu'il les met bout-à-bout), avec une taille maximale de 520 octets pour le résultat. Cet opcode était présent dans Bitcoin à l'origine, mais a été désactivé par Satoshi Nakamoto en 2010 par précaution, en raison de risques potentiels de déni de service liés à la croissance exponentielle des données sur la pile. Sa réactivation dans Tapscript permettrait de construire des covenants, des arbres de Merkle vérifiables en Script, des schémas de signature de Lamport résistants aux ordinateurs quantiques, et de nombreuses autres primitives cryptographiques. Le BIP-0347, proposé par Ethan Heilman et Armin Sabouri, est l'une des propositions de soft fork les plus débattues depuis l'activation de Taproot, en raison de la puissance et de la polyvalence que cet opcode simple pourrait offrir.

Termes associés : **OP_CAT - 0X7E** **TAPSCRIPT**

BIP-0348

BIP

Propose l'ajout de l'opcode `OP_CHECKSIGFROMSTACK` pour Tapscript, en remplaçant `OP_SUCCESS204`. Contrairement à `OP_CHECKSIG` qui vérifie une signature sur le hash de la transaction courante, cet opcode permet de vérifier une signature Schnorr (BIP-0340) sur un message arbitraire fourni sur la pile. Il consomme trois éléments : une clé publique de 32 octets, un message et une signature. Le BIP ne définit volontairement pas de variante VERIFY, par souci de simplicité. Combiné avec `OP_TXHASH` (BIP-0346), il permet une construction de hash de signature entièrement généralisée, ouvrant la voie à l'introspection de transaction, à la délégation de droits de dépense, et à la mise en place de canaux Lightning Symmetry (LN-Symmetry / Eltoo). Proposé par Brandon Black et Jeremy Rubin, son statut est « Draft ».

Termes associés : **COVENANT** **OPCODES**

BIP-0349

BIP

Proposition de Brandon Black et Jeremy Rubin introduisant l'opcode `OP_INTERNALKEY` dans Tapscript. Cet opcode remplace `OP_SUCCESS203` (`0×cb`) et place la représentation *x-only* de 32 octets de la clé interne Taproot (*internal key*) directement sur la pile lors de l'exécution d'un script. Actuellement, un tapscript qui a besoin de référencer la clé interne doit l'inclure explicitement dans ses données, ce qui consomme 8 vBytes supplémentaires on-chain. `OP_INTERNALKEY` élimine cette redondance.

Au-delà de cette économie d'espace, l'opcode ouvre un cas d'usage important : le « *re-keying* » avec préservation de l'arbre de Merkle. Lorsque la clé interne d'une sortie Taproot est modifiée (par exemple, pour ajouter ou retirer un participant), les scripts dans les feuilles de l'arbre qui référencent cette clé en dur deviennent incohérents. Avec `OP_INTERNALKEY`, les feuilles se re-keyent automatiquement, car l'opcode renvoie toujours la clé interne courante.

Termes associés : **OPCODES** **TAPROOT**

BIP-0350

BIP

Introduit Bech32m, une variante du format d'adresse Bech32 (BIP-0173) qui corrige une faiblesse dans la détection d'erreurs pour les adresses SegWit v1 et supérieures. Le format Bech32 original présentait un défaut : l'insertion ou la suppression de caractères « q » juste avant un caractère « p » final n'était pas toujours détectée par la somme de contrôle. Bech32m résout ce problème en modifiant la constante finale de la *checksum*. Les adresses SegWit v0 (commençant par `bc1q`) conservent le format Bech32 original, tandis que les adresses v1 et supérieures (comme `bc1p` pour Taproot) utilisent Bech32m.

Termes associés : **BIP-0173** **BIP-0341**

BIP-0351

BIP

Proposition d'un protocole de paiements privés utilisant un code de paiement publié par le destinataire sur le réseau. Le payeur dérive une adresse unique pour le destinataire sans interaction préalable, en utilisant une transaction de notification chiffrée. Le BIP-0351 constitue une approche alternative aux *silent payments* (BIP-0352), avec des compromis différents : il nécessite une transaction de notification *on-chain*, mais offre une compatibilité plus large avec les portefeuilles existants. Les deux propositions visent à éliminer la réutilisation d'adresses tout en préservant la confidentialité du destinataire.

Termes associés : **PAYJOIN** **SILENT PAYMENTS**

BIP-0352

BIP

Proposition d'amélioration de Josibake et Ruben Somsen qui introduit les Silent Payments, une méthode pour utiliser des adresses Bitcoin statiques afin de recevoir des paiements sans pour autant produire de réutilisation d'adresse, sans interaction et sans lien visible on-chain entre les différents paiements. Cette technique élimine le besoin de générer de nouvelles adresses de réception vierges pour chaque transaction, ce qui permet d'éviter les interactions habituelles dans Bitcoin où le destinataire doit fournir une nouvelle adresse au payeur.

Dans ce système, le payeur utilise la clé publique du destinataire et sa clé privée personnelle pour générer une adresse vierge pour chaque paiement. Seul le destinataire, grâce à sa clé privée, peut calculer la clé privée correspondant à cette adresse. On utilise ECDH (*Elliptic-Curve Diffie-Hellman*), un algorithme cryptographique d'échange de clés, pour établir un secret partagé qui sert ensuite à dériver l'adresse de réception et la clé privée (uniquement du côté du destinataire). Pour identifier les Silent Payments qui leur sont destinés, les destinataires doivent scanner la blockchain et examiner chaque transaction correspondant aux critères des Silent Payments. Contrairement au BIP-0047, qui utilise une transaction de notification pour établir le canal de paiement, les Silent Payments suppriment cette étape, ce qui permet d'économiser une transaction. Toutefois, le compromis est que le destinataire doit scanner l'ensemble des transactions potentielles pour déterminer, en appliquant ECDH, si elles lui sont adressées.

Termes associés : **BIP-0047** **ECDH**

BIP-0353

BIP

Proposition d'amélioration de Bitcoin qui définit un système de résolution d'adresses de paiement lisibles par un humain, basé sur le DNS. Le BIP-0353 permet d'associer un identifiant au format `utilisateur@domaine.com` à des instructions de paiement Bitcoin, en stockant ces informations dans des enregistrements DNS de type TXT, sécurisés par DNSSEC.

Concrètement, lorsqu'un portefeuille reçoit une adresse comme `satoshi@example.com`, il effectue une requête DNS vers :

```
satoshi.user._bitcoin-payment.example.com
```

Ça lui permet de récupérer un URI de paiement au format BIP-0021. Cet URI peut contenir une adresse *on-chain*, un *offer* BOLT-12 ou des instructions pour les *silent payments*, ce qui rend le mécanisme compatible avec différents protocoles de paiement.

L'avantage principal est l'expérience utilisateur : les adresses deviennent mémorisables et réutilisables, sans sacrifier la confidentialité puisque les protocoles sous-jacents (BOLT-12, *silent payments*...) génèrent des adresses/*invoices* uniques pour chaque paiement. Le BIP-0353 a été proposé

par Matt Corallo et Bastien Teinturier en 2024.

Termes associés : **BOLT-12** **SILENT PAYMENTS**

BIP-0360

BIP

Propose un nouveau type de sortie appelé « Pay-to-Merkle-Root » (P2MR), via un soft fork. P2MR fonctionne de manière similaire à P2TR (Taproot, BIP-0341), mais en supprimant le chemin de dépense par clé (*key-path spend*). Cette modification vise principalement à protéger les fonds contre les attaques à long terme par des ordinateurs quantiques, qui pourraient utiliser l'algorithme de Shor (ou un autre algorithme encore inconnu) pour dériver des clés privées à partir de clés publiques exposées sur la blockchain. En retirant la clé publique interne de la sortie, P2MR empêche cette exposition et permet d'utiliser les scripts Tapscript de manière résistante aux attaques quantiques à long terme.

Termes associés : **OPCODES** **TAPROOT**

BIP-0370

BIP

Définit la seconde version du format PSBT (BIP-0174). Contrairement à la version 0, où la transaction non signée était figée dès la création, le PSBT v2 permet la modification interactive de la transaction avant la signature : les participants peuvent ajouter ou supprimer des inputs et des outputs de manière collaborative.

Termes associés : **BIP-0174** **PSBT**

BIP-0371

BIP

Ajoute des champs spécifiques à Taproot dans le format PSBT (BIP-0174). Ces champs incluent la clé interne (*internal key*), le chemin de Merkle (*Merkle path*) des scripts, les signatures de Schnorr, et les feuilles de script Taproot (*tap leaf scripts*). Cette extension est nécessaire pour que les hardware wallets et les coordinateurs multisig puissent correctement construire, vérifier et signer des transactions Taproot via le format PSBT.

Termes associés : **BIP-0174** **BIP-0341**

BIP-0372

BIP

Ajout de champs PSBT (BIP-0174) dédiés aux tweaks *pay-to-contract*. Le BIP-0372 permet aux signataires de savoir qu'une clé publique a été modifiée par un tweak P2C et de signer correctement la transaction en tenant compte de cette modification.

Termes associés : **BIP-0174** **TWEAK**

BIP-0373

BIP

Ajoute des champs spécifiques à MuSig2 (BIP-0327) dans le format PSBT afin de permettre la signature collaborative via des *hardware wallets* et des coordinateurs. Ces champs incluent les clés publiques des participants, les *nonces* publics et les signatures partielles. Le BIP-0373 permet ainsi de coordonner le protocole MuSig2 en deux tours par l'échange de fichiers PSBT, sans que les différents signataires aient besoin d'interagir directement entre eux.

Termes associés : **BIP-0174** **BIP-0327**

BIP-0374

BIP

Définit un protocole de preuve d'égalité de logarithmes discrets (*DLEQ proof*) pour la courbe secp256k1. Ce mécanisme cryptographique permettrait de prouver que deux points sur la courbe elliptique partagent le même scalaire secret, sans révéler la valeur de ce scalaire.

Termes associés : **BIP-0327** **BIP-0352**

BIP-0375

BIP

Définit les champs PSBT nécessaires pour envoyer des Silent Payments (BIP-0352). Ce BIP inclut les *shares* ECDH de chaque signataire ainsi que les preuves DLEQ (BIP-0374) attestant de l'exactitude des calculs. Le BIP-0375 permet à plusieurs signataires dans un contexte multisig de collaborer pour calculer les adresses de sortie des Silent Payments via l'échange de fichiers PSBT, sans que chaque participant ait besoin de révéler sa clé privée aux autres.

Termes associés : **BIP-0174** **BIP-0352**

BIP-0379

BIP

Proposition formalisant Miniscript, une représentation structurée d'un sous-ensemble de Bitcoin Script qui permet l'analyse, la composition et la signature générique de conditions de dépense. Miniscript établit une correspondance entre les opcodes de Script et un système de types composables, où les conditions de dépense peuvent être combinées à l'aide d'opérateurs logiques (AND, OR, seuils) avec des garanties de correction.

Cette approche permet aux portefeuilles de déterminer automatiquement les exigences de signature pour une sortie donnée, d'estimer précisément les frais de transaction en calculant la taille maximale du témoin, et de vérifier qu'une politique de dépense est satisfaisable. Miniscript a été développé par Pieter Wuille, Andrew Poelstra et Sanket Kanjalkar.

Termes associés : **MINISCRIPT** **SCRIPT**

BIP-0380

BIP

Proposition d'amélioration qui introduit un langage standard pour décrire les collections de scripts de sortie (*output scripts*) des portefeuilles HD Bitcoin. Ce langage est appelé « *Output Script Descriptors* ». Il vise à standardiser la manière de représenter et de gérer les scripts de sortie, afin de faciliter la sauvegarde, l'exportation et l'importation de portefeuilles. En plus des données privées comme la phrase de récupération, les *descriptors* fournissent toutes les informations nécessaires pour retrouver les paires de clés utilisées dans un portefeuille HD. Le BIP-0380 décrit le fonctionnement général des *descriptors*, tandis que les BIP-0381, BIP-0382, BIP-0383, BIP-0384, BIP-0385 et BIP-0386 spécifient les expressions utilisées. Le BIP-0380 a été implémenté avec tous les autres BIP liés aux *descriptors* (sauf le BIP-0386) dans la version 0.17 de Bitcoin Core.

Terme associé : **OUTPUT SCRIPT DESCRIPTORS**

BIP-0381

BIP

Introduit des fonctions pour les descriptors, notamment `pk(KEY)` (Pay-to-PubKey), `pkh(KEY)` (Pay-to-PubKey-Hash), et `sh(SCRIPT)` (Pay-to-Script-Hash). Ces fonctions standardisent la manière de décrire les types de scripts Legacy dans les descriptors. Le BIP-0381 a été implémenté avec tous les autres BIP liés aux descriptors (sauf le BIP-0386) dans la version 0.17 de Bitcoin Core.

Terme associé : **OUTPUT SCRIPT DESCRIPTORS**

BIP-0382

BIP

Introduit les fonctions `wpkh(KEY)` (*Pay-to-Witness-PubKey-Hash*) et `wsh(SCRIPT)` (*Pay-to-Witness-Script-Hash*) pour les *descriptors*. Ces fonctions standardisent la manière de décrire les types de scripts SegWit dans les *descriptors*. Le BIP-0382 a été implémenté avec tous les autres BIP liés aux *descriptors* (sauf le BIP-0386) dans la version 0.17 de Bitcoin Core.

Terme associé : **OUTPUT SCRIPT DESCRIPTORS**

BIP-0383

BIP

Introduit les fonctions `multi(NUM, KEY, ..., KEY)` et `sortedmulti(NUM, KEY, ..., KEY)` pour les descriptors. Ces fonctions permettent de décrire les scripts multisignatures dans les descriptors, avec `sortedmulti` qui trie les clés publiques par ordre lexicographique pour assurer la compatibilité lors de l'import. Le BIP-0383 a été implémenté avec tous les autres BIP liés aux descriptors (sauf le BIP-0386) dans la version 0.17 de Bitcoin Core.

Terme associé : **OUTPUT SCRIPT DESCRIPTORS**

BIP-0384

BIP

Introduit la fonction `combo(KEY)` pour les descriptors. Cette fonction décrit un ensemble de scripts de sortie possibles pour une clé publique donnée. Il permet donc de couvrir plusieurs types de scripts en même temps, notamment P2PK, P2PKH, P2WPKH et P2SH-P2WPKH. Si la clé donnée est compressée, les 4 types de scripts sont testés, et si elle ne l'est pas, seuls les 2 types de scripts Legacy sont testés. L'objectif est de simplifier la représentation des clés dans les descriptors en fournissant une méthode unique pour générer différentes variantes de scripts de sortie établies sur une même clé publique. Le BIP-0384 a été implémenté avec tous les autres BIP liés aux descriptors (sauf le BIP-0386) dans la version 0.17 de Bitcoin Core.

Terme associé : **OUTPUT SCRIPT DESCRIPTORS**

BIP-0385

BIP

Introduit les fonctions de descriptors `addr(ADDR)` et `raw(HEX)`. La fonction `addr(ADDR)` permet de décrire un script de sortie en utilisant une adresse de réception, tandis que la fonction `raw(HEX)` permet de spécifier un script de sortie en utilisant une représentation hexadécimale brute de ce script. Le BIP-0385 a été implémenté avec tous les autres BIP liés aux descriptors (sauf le BIP-0386) dans la version 0.17 de Bitcoin Core.

Terme associé : **OUTPUT SCRIPT DESCRIPTORS**

BIP-0386

BIP

Introduit les fonctions de descriptors pour Taproot. Il définit les fonctions `tr(KEY)` et `tr(KEY, TREE)` pour trouver des sorties Taproot, où `KEY` est la clé interne et `TREE` est une arborescence optionnelle de chemins de script. Le BIP-0386 a été implémenté dans la version 22.0 de Bitcoin Core.

Terme associé : **OUTPUT SCRIPT DESCRIPTORS**

BIP-0387

BIP

Proposition définissant les output script descriptors `multi_a()` et `sortedmulti_a()` pour le multisig en Tapscript utilisant l'opcode `OP_CHECKSIGADD`. Ces descriptors sont l'équivalent de `multi()` et `sortedmulti()` utilisés dans les scripts traditionnels, mais adaptés à la dépense par chemin de script Taproot (*script-path spending*). La variante `sortedmulti_a()` trie les clés publiques dans l'ordre lexicographique pour garantir un résultat déterministe.

Termes associés : **OUTPUT SCRIPT DESCRIPTORS** **TAPSCRIPT**

BIP-0388

BIP

Proposition définissant les « politiques de portefeuille » (*wallet policies*), un sous-ensemble restreint d'output script descriptors optimisé pour les portefeuilles matériels. Les descripteurs complets peuvent être arbitrairement complexes et difficiles à afficher sur les écrans limités des appareils de signature. Le BIP-0388 simplifie le format en séparant la structure du descripteur (le « template ») de la liste des clés, tout en restant suffisamment expressif pour couvrir les configurations courantes : singlesig, multisig et MuSig2. Cette approche permet à l'utilisateur de vérifier visuellement la politique de dépense sur l'écran de son hardware wallet avant de signer une transaction.

Termes associés : **BIP-0032** **OUTPUT SCRIPT DESCRIPTORS**

BIP-0389

BIP

Proposition étendant les output script descriptors avec des expressions de clé multichemin (*multipath key expressions*) utilisant la syntaxe `<M;N>`. Par exemple, `<0;1>` dans un chemin de dérivation permet de représenter à la fois les adresses de réception (index 0) et les adresses de change (index 1) au sein d'un même descripteur. Sans cette extension, il faudrait deux descripteurs distincts pour un même portefeuille. Le BIP-0389 simplifie ainsi la sauvegarde d'un portefeuille en la réduisant à une seule chaîne de caractères contenant toute l'information nécessaire à la reconstruction des adresses.

Termes associés : **BIP-0032** **OUTPUT SCRIPT DESCRIPTORS**

BIP-0390

BIP

Proposition définissant l'expression de clé `musig()` pour les output script descriptors. Cette expression permet de représenter une clé agrégée MuSig2 au sein d'un descripteur, rendant possible la description standardisée d'une dépense Taproot par chemin de clé (*key-path*) impliquant plusieurs signataires. Sans cette expression, il n'existait pas de moyen normalisé de décrire un multisig MuSig2 dans les descripteurs.

Termes associés : **MULTISIG** **OUTPUT SCRIPT DESCRIPTORS**

BIP-0431

BIP

BIP informationnel qui décrit les attaques de « *transaction pinning* », et propose des restrictions topologiques sur les paquets de transactions dans la mempool afin d'atténuer ce type d'attaque. Le pinning est une technique où un attaquant exploite les limitations des politiques de mempool pour empêcher la confirmation d'une transaction légitime, ce qui pose problème notamment pour les protocoles de couche supérieure comme Lightning.

Le BIP-0431 propose de limiter la structure des paquets de transactions : pas de parents non confirmés multiples, taille de paquet restreinte, et interdiction de certaines topologies complexes. Ces contraintes rendent le remplacement de transactions plus prévisible et fiable, facilitant ainsi le fonctionnement des protocoles qui dépendent de la confirmation rapide de transactions.

Termes associés : **MEMPOOL** | **POLITIQUE DE MEMPOOL**

BIP-0433

BIP

BIP informationnel qui définit un nouveau type de sortie standard appelé « Pay-to-Anchor » (P2A). Il s'agit d'un output SegWit version 1 utilisant un script de 2 octets (`OP_1 0×4e73`) que n'importe qui peut dépenser. Ce type de sortie est spécifiquement conçu pour servir d'ancrage au mécanisme CPFP (*Child Pays For Parent*) : tout participant peut créer une transaction enfant dépensant cet output pour augmenter les frais effectifs de la transaction parente. Le P2A remplace les *anchor outputs* spécifiques à Lightning par un mécanisme universel, utilisable par tout protocole nécessitant un ajustement de frais après la création d'une transaction pré-signée.

Termes associés : **ANCHOR OUTPUTS** | **CPFP - CHILD PAY FOR PARENT**

BIP-0434

BIP

Propose un mécanisme générique de négociation de fonctionnalités entre pairs du réseau Bitcoin lors de l'établissement d'une connexion. Actuellement, chaque nouvelle fonctionnalité du protocole P2P nécessite l'ajout de messages spécifiques échangés après le handshake de version. Le BIP-0434 remplace cette approche ad hoc par un système unifié où les nœuds annoncent les fonctionnalités qu'ils supportent via un ensemble standardisé de messages de négociation. Cela simplifie l'implémentation de nouvelles fonctionnalités réseau et réduit la complexité du protocole de connexion entre pairs.

Termes associés : **POLITIQUE DE MEMPOOL** | **RÉSEAU BITCOIN**

BIP-0443

BIP

BIP rédigé par Salvatore Ingala qui propose l'ajout de `OP_CHECKCONTRACTVERIFY` (CCV), un opcode généraliste pour implémenter des covenants sur Bitcoin. CCV permet de vérifier qu'une sortie de transaction respecte un contrat défini par une combinaison d'un hash et de données embarquées dans la clé publique Taproot via le mécanisme de *tweaking*. Contrairement aux propositions précédentes ciblant un cas d'usage spécifique, CCV offre un cadre flexible pouvant servir à de multiples constructions : vaults (coffres-forts) avec mécanisme de récupération, canaux Lightning améliorés, pools de paiement, et autres protocoles nécessitant une introspection des sorties de transaction.

Termes associés : **COVENANT** **OPCODES**

BIRTH DATE

PORTEFEUILLE | FR : DATE DE NAISSANCE

Fait référence à la date à laquelle un portefeuille a été créé. Cette information est importante lors de la restauration d'un portefeuille, car elle permet au logiciel de savoir à partir de quel bloc il doit commencer à chercher les transactions associées à ce portefeuille. En connaissant la date de naissance, le logiciel peut efficacement synchroniser et récupérer l'historique des transactions sans avoir à analyser l'ensemble de la blockchain depuis le bloc de Genèse.

Termes associés : **GAP LIMIT** **GRAINE**

BISQ

OUTIL

Logiciel open source de plateforme d'échange décentralisée (DEX) permettant d'acheter et de vendre des bitcoins en pair-à-pair, sans intermédiaire centralisé ni vérification d'identité (KYC). Bisq fonctionne comme une application de bureau qui se connecte à un réseau P2P sur Tor. Chaque échange est sécurisé par un système de dépôt de garantie en bitcoins (multisig 2-de-2) et un processus d'arbitrage en cas de litige. Les échanges se font contre des monnaies fiat (virement bancaire, Revolut, etc.) ou d'autres cryptomonnaies. Bisq ne détient jamais les fonds des utilisateurs et ne collecte aucune donnée personnelle. Le logiciel est financé par une DAO (organisation autonome décentralisée) intégrée qui utilise le token BSQ pour sa gouvernance.

Termes associés : **NOEUD** **PORTEFEUILLE**

BIT

</> INFORMATIQUE | FR : CHIFFRE BINAIRE

Le mot « bit » est la contraction des termes « binary » et « digit » en anglais. Dans le contexte de l'informatique et de la cryptologie, un bit est l'unité fondamentale d'information numérique et représente la plus petite quantité d'informations possible. Un bit ne peut prendre que deux valeurs distinctes : `0` ou `1`. Ces valeurs sont également appelées états binaires et peuvent représenter diverses choses, telles que les réponses `oui` ou `non`, `vrai` ou `faux` et `on` ou `off`. Les bits sont la base des systèmes numériques et sont utilisés pour stocker et transmettre de l'information dans les ordinateurs et les réseaux. Le nom de « Bitcoin » provient sûrement de la concaténation du terme « Bit », pour évoquer la nature électronique du système de paiement, et du terme « Coin », pour évoquer son objectif monétaire.

Termes associés : **BINAIRE** **OCTET**

BIT GOLD

HISTOIRE

Système de monnaie numérique décentralisée conceptualisé par Nick Szabo en 1998 et décrit publiquement en 2005. Bit gold a été conçu pour générer et échanger une ressource virtuelle appelée le bit gold. Ce système ne reposait sur aucun bien physique, mais visait à créer une forme de rareté infalsifiable. Le protocole bit gold reposait sur la création monétaire par preuve de travail, où les morceaux de bit gold étaient créés via la puissance de calcul des ordinateurs, formant ainsi une chaîne de preuve de travail. Chaque preuve de travail était horodatée puis ajoutée à un registre de propriété. La vérification et le transfert de la propriété de bit gold étaient effectués via un registre public, où les utilisateurs étaient identifiés par des clés publiques. Bit gold est resté à l'état de concept et n'a jamais été implémenté. Ce système est un des précurseurs de Bitcoin avec b-money et RPoW, mais Satoshi semblait ne pas connaître son existence avant la création de Bitcoin. Il y a fait mention plus tard sur le site et sur le forum de Bitcoin.

Termes associés : **B-MONEY** **RPOW**

BIT - UNITÉ

$ ÉCONOMIE ET RÉGULATION

Dans le contexte de Bitcoin, le terme « bit » est aussi utilisé pour désigner une subdivision monétaire du bitcoin. Un bit est égal à 100 satoshis, qui représentent la plus petite unité indivisible de bitcoin. Ainsi, un bitcoin est égal à 1 000 000 de bits ou 100 000 000 de satoshis. Cependant, l'utilisation de ce terme comme subdivision monétaire est sujette à controverse. La majorité des bitcoiners emploient soit le « sats », soit le « btc », mais pas le « bit ».

Termes associés : **BTC** **XBT**

BITAXE

↗ MINAGE

Projet open source de mineur de Bitcoin miniature basé sur une puce ASIC. Le Bitaxe est un petit appareil de minage conçu pour être accessible, peu coûteux et entièrement transparent dans sa conception, avec des schémas matériels et un firmware publiés en open source. Il utilise des puces ASIC de minage (issues de machines grand public comme les séries Antminer de Bitmain) soudées sur un circuit imprimé compact et autonome, alimenté par un simple câble USB-C. Le Bitaxe est principalement utilisé à des fins éducatives ou pour le « lottery mining » (minage solo en espérant trouver un bloc de façon probabiliste), car sa puissance de calcul reste très faible comparée aux grandes fermes de minage industrielles.

Termes associés : **ASIC** | **MINAGE**

BITBANANA

✱ OUTIL

BitBanana est une application Android open source permettant de gérer à distance un nœud Lightning depuis son téléphone. Né comme un fork du portefeuille Zap Android (créé par Jack Mallers), le projet a été repris et renommé par le développeur Michael Wünsch pour poursuivre un développement indépendant. L'application offre une interface complète pour superviser et piloter son nœud : gestion des canaux, envoi et réception de paiements, consultation des logs et des statistiques de routage. BitBanana est compatible avec LND, Core Lightning et Nostr Wallet Connect, ce qui en fait un outil polyvalent pour les opérateurs de nœuds Lightning.

Termes associés : **LND** | **NOEUD LIGHTNING**

BITBOX

⊸ PORTEFEUILLE

Hardware wallet fabriqué par la société suisse Shift Crypto. Le modèle actuel, le BitBox02, existe en deux versions : une édition multi-crypto et une édition Bitcoin-only dont le firmware est strictement limité à Bitcoin, réduisant ainsi la surface d'attaque. L'appareil intègre un secure element, un firmware open source et une interface tactile par glissement sur les côtés de l'appareil.

Termes associés : **HARDWARE WALLET** | **MULTISIG**

BITCOIN ATM

$ ÉCONOMIE ET RÉGULATION | FR : DISTRIBUTEUR DE BITCOINS

Borne physique qui permet d'acheter ou, dans certains cas, de vendre des bitcoins en échange de monnaie fiduciaire (espèces ou carte bancaire). Aussi appelé BTM (*Bitcoin Teller Machine*), ce type de distributeur offre un point d'accès direct à Bitcoin sans passer par une plateforme d'échange en ligne.

Termes associés : **AML - ANTI-MONEY LAUNDERING** | **KYC - KNOW YOUR CUSTOMER**

BITCOIN - B MAJUSCULE

PROTOCOLE

Bitcoin est le nom du système de cash électronique pair-à-pair créé par Satoshi Nakamoto en 2009. L'utilisation du terme Bitcoin avec un « B » majuscule peut vouloir évoquer trois choses différentes :

- Le système Bitcoin ;
- Le protocole Bitcoin ;
- Le réseau Bitcoin.

Le terme de bitcoin avec un « b » minuscule est généralement réservé pour évoquer l'unité monétaire échangée sur ce système.

Termes associés : BITCOIN - B MINUSCULE | BTC

BITCOIN - B MINUSCULE

PROTOCOLE

Le bitcoin (écrit avec un « b » minuscule) fait référence à l'unité monétaire utilisée pour les échanges sur le système de cash électronique Bitcoin (avec un « B » majuscule). Le bitcoin est souvent abrégé en « BTC » ou « XBT » et sert de moyen d'échange, de réserve de valeur et d'unité de compte au sein du réseau. Chaque bitcoin est divisible en 100 millions d'unités plus petites, appelées « satoshis » ou « sats », en l'honneur de son créateur, Satoshi Nakamoto. Les bitcoins sont émis par le processus de la preuve de travail (minage). Le nombre total de bitcoins est limité à environ 21 millions.

Termes associés : BITCOIN - B MAJUSCULE | BTC

BITCOIN BEACH

COMMUNAUTÉ

Initiative communautaire lancée dans le village d'El Zonte, sur la côte pacifique du Salvador, qui a transformé cette petite localité en la première économie circulaire fonctionnant sur Bitcoin. Le projet est né en 2019 grâce au don d'un bitcoiner resté anonyme, avec la volonté de prouver que Bitcoin pouvait améliorer concrètement le quotidien de populations exclues du système bancaire traditionnel.

Bitcoin Beach a permis aux habitants de payer leurs factures (eau, électricité, téléphone), leurs courses alimentaires ou même le coiffeur directement en bitcoins via le Lightning Network. Le portefeuille développé pour le projet, initialement appelé « Bitcoin Beach Wallet », a ensuite été renommé « Blink ».

L'expérience d'El Zonte a joué un rôle important dans la décision du Salvador d'adopter Bitcoin comme monnaie à cours légal en 2021, faisant du pays le premier État à franchir ce pas. Le modèle a depuis inspiré des initiatives similaires dans d'autres pays, notamment Bitcoin Jungle au Costa Rica.

Terme associé : LIGHTNING NETWORK

BITCOIN CLASSIC

HISTOIRE

Proposition de fork du protocole lancée en janvier 2016 à la suite de l'abandon de Bitcoin XT. Elle visait à augmenter la limite de taille des blocs de 1 Mo à 2 Mo avec le BIP-0109, une approche jugée plus modérée par rapport à Bitcoin XT qui proposait une augmentation beaucoup plus élevée (8 Mo + doublement tous les 2 ans). Cette initiative était menée par les développeurs Gavin Andresen et Jeff Garzik, et bénéficiait de l'appui de grandes entreprises de l'écosystème comme Coinbase. Bitcoin Classic est apparu dans un contexte de forte intensification de la Blocksize War et était classé parmi le camp des big blockers. Bien que Bitcoin Classic ait attiré l'attention et le soutien de nombreux acteurs du secteur, son mécanisme d'activation, qui nécessitait un soutien de 75 % des mineurs, était controversé et a finalement contribué à son échec.

Terme associé : **BLOCKSIZE WAR**

BITCOIN-CLI

PROTOCOLE

`Bitcoin-cli`, acronyme pour « *Bitcoin Command Line Interface* », est une interface de ligne de commande conçue pour interagir avec une instance de Bitcoin Core en exécution, en particulier le daemon, `bitcoind`. Il s'agit d'un programme indépendant qui offre à l'utilisateur un moyen de communiquer et d'exécuter des commandes pour contrôler et interroger l'état de l'instance de `bitcoind`. En plus des capacités de gestion du réseau, telles que la surveillance des transactions et des blocs, `bitcoin-cli` offre également des fonctionnalités de portefeuille, permettant aux utilisateurs d'effectuer des transactions Bitcoin en envoyant et recevant des fonds.

Termes associés : **BITCOIN CORE** **BITCOIND**

BITCOIN.CONF

PROTOCOLE

Fichier de configuration utilisé pour personnaliser le fonctionnement d'un nœud Bitcoin exécutant le client Bitcoin Core. Situé dans le répertoire de données de Bitcoin Core, ce fichier texte permet aux opérateurs de nœuds de spécifier divers paramètres et options qui influencent le comportement du nœud. Parmi les paramètres que l'on peut définir dans `bitcoin.conf`, on trouve des éléments tels que la taille de la Mempool, les restrictions sur les connexions réseau, les frais de transaction minimum de relai, ainsi que d'autres options de sécurité et de performances.

Termes associés : **BITCOIND** **MEMPOOL**

BITCOIN CORE

PROTOCOLE

Bitcoin Core est le logiciel *open-source* de référence pour le système Bitcoin, et constitue la principale implémentation du protocole Bitcoin à ce jour. Il est développé et maintenu par un large groupe de contributeurs bénévoles. Initialement nommé « Bitcoin Qt », c'est le troisième client de l'histoire de Bitcoin après Bitcoin, de Satoshi Nakamoto, et Bitcoind. Il a été développé à partir du code original de Satoshi et a introduit une interface graphique pour l'utilisateur. Par ailleurs, encore aujourd'hui, l'interface graphique de Bitcoin Core s'appelle `bitcoin-qt`. Il est fourni avec `bitcoind` depuis la version 0.5. Le logiciel Bitcoin Core sert à plusieurs fins. Tout d'abord, il agit comme un client nœud complet. Bitcoin Core inclut également un portefeuille (*wallet*) pour les utilisateurs qui souhaitent stocker, gérer et effectuer des transactions directement avec Bitcoin Core.

Termes associés : **BITCOIN QT** **BITCOIND**

BITCOIN CORE GUI-QML

OUTIL

Projet en cours de création d'une application de portefeuille et de nœud Bitcoin établie sur Bitcoin Core, mais avec une interface graphique intuitive développée à l'aide de QML (*Qt Modeling Language*). L'objectif de Bitcoin Core GUI-QML est de moderniser l'interface utilisateur de Bitcoin Core pour offrir une expérience plus simple, plus fluide et plus dynamique.

L'interface traditionnelle de Bitcoin Core est principalement développée en C++ avec Qt Widgets, qui est un système qui produit des interfaces utilisateur très classiques. Ce nouveau projet s'appuie dorénavant sur QML, qui est un langage déclaratif utilisé pour concevoir des GUI avec des animations, des transitions et une interface plus réactive et stylisée. Cela permettra à Core de se rapprocher des interfaces d'applications modernes avec des interactions riches. L'objectif est que ce logiciel soit disponible à la fois sur PC et sur mobiles.

Termes associés : **BITCOIN CORE** **BITCOIN QT**

BITCOIN-DEV

PROTOCOLE

Liste de diffusion d'e-mail dédiée au développement du protocole Bitcoin. Bitcoin-dev a été créée par Jeff Garzik en 2011 et initialement hébergée sur Sourceforge.net. La liste de diffusion a ensuite été déplacée sur Linux Foundation, puis sur Open Source Lab, et enfin, depuis février 2024, elle est hébergée sur Google Groups : https://groups.google.com/g/bitcoindev

Cette liste de diffusion représente un point d'échange pour les développeurs et les chercheurs impliqués dans l'évolution du Bitcoin. Les discussions y sont strictement techniques ou académiques. C'est notamment ici que doivent être envoyés les brouillons de BIPs (proposition d'amélioration de Bitcoin), avant d'être éventuellement intégrés au dépôt GitHub des BIPs. Ses modérateurs sont actuellement Ruben Somsen, Bryan Bishop et Warren Togami.

La participation à cette liste est ouverte à tous les contributeurs, mais elle s'adresse principalement à ceux qui ont une expertise technique dans le domaine de Bitcoin. Si vous souhaitez recevoir ces échanges par e-mail, vous pouvez vous abonner à cette liste de diffusion. Si vous souhaitez plutôt avoir un résumé de ces discussions, vous pouvez également vous abonner à la newsletter de Bitcoin Optech : https://bitcoinops.org/en/newsletters/

Termes associés : **BIP** | **BITCOIN OPTECH**

BITCOIN FOG

CONFIDENTIALITÉ

Service de mixage centralisé qui a opéré de 2011 à 2021. Bitcoin Fog offrait aux utilisateurs la possibilité d'accroître leur confidentialité en mélangeant leurs bitcoins avec ceux d'autres utilisateurs, dans le but de dissocier les pièces de leur historique de transactions. Étant donné sa nature centralisée, les utilisateurs devaient faire confiance à l'opérateur du service pour ne pas détourner les fonds et pour ne pas conserver de trace des opérations de mixage.

Termes associés : **COINJOIN** | **MÉLANGEUR**

BITCOIN INQUISITION

PROTOCOLE

Fork logiciel de Bitcoin Core qui vise à tester l'intégration de nouvelles propositions d'amélioration de Bitcoin dans un environnement contrôlé sur des réseaux de test comme un Signet. Il inclut, par exemple, le support pour le BIP-0118 (ANYPREVOUT) et le BIP-0119 (CHECKTEMPLATEVERIFY). Le but de Bitcoin Inquisition est d'expérimenter, afin d'avoir une meilleure compréhension des applications, des avantages, des risques et des compromis liés à ces modifications. Bitcoin Inquisition est maintenu par Anthony Towns.

Termes associés : **BITCOIN CORE** | **SIGNET**

BITCOIN JESUS

HISTOIRE

Surnom donné à l'entrepreneur et investisseur Roger Ver, qui a été un promoteur de Bitcoin dès le début des années 2010. Ce surnom lui a été donné en raison de son rôle influent dans la popularisation de Bitcoin à son lancement. Il est également connu pour son soutien au *hard fork* Bitcoin Cash (BCH), qu'il considère comme plus fidèle à la vision originale de Bitcoin.

Termes associés : **BCH - BITCOIN CASH** | **BLOCKSIZE WAR**

BITCOIN KEEPER

PORTEFEUILLE

Portefeuille Bitcoin open source développé par BitHyve, spécialisé dans le multisig. L'application permet de créer des coffres (*vaults*) multisignatures en combinant différents portefeuilles matériels et clés logicielles. Bitcoin Keeper prend en charge le BIP-0085 pour la dérivation de sous-portefeuilles, l'achat de bitcoins directement en stockage à froid, et le transfert automatique vers les coffres multisig. Il est disponible sur mobile (iOS, Android) et sur ordinateur.

Termes associés : **HARDWARE WALLET** | **MULTISIG**

BITCOIN KNOTS

PROTOCOLE

Implémentation minoritaire du protocole Bitcoin. Bitcoin Knots est une alternative au logiciel de référence Bitcoin Core, proposant quelques règles et fonctionnalités différentes, tout en étant compatible avec les autres nœuds. Knots est développé et maintenu par Luke Dashjr.

Termes associés : **BITCOIN CORE** | **NOEUD**

BITCOIN MAGAZINE

COMMUNAUTÉ

Publication fondée en 2012 par Mihai Alisie et Vitalik Buterin (avant que ce dernier ne crée Ethereum), dédiée à l'actualité et à l'analyse de Bitcoin. Bitcoin Magazine est l'un des plus anciens médias spécialisés dans l'écosystème Bitcoin. Il propose des articles, des analyses techniques, des interviews et des contenus éducatifs couvrant l'ensemble des aspects de Bitcoin : protocole, économie, régulation, minage et développement. Le magazine existe sous forme de site web, de publication papier et de contenus vidéo. Bitcoin Magazine organise également la « Bitcoin Conference », l'un des plus grands événements annuels de la communauté Bitcoin, qui se tient aux États-Unis et rassemble des développeurs, entrepreneurs, investisseurs et enthousiastes du monde entier.

Terme associé : **HALVING**

BITCOIN OPTECH

COMMUNAUTÉ

Organisation à but non lucratif qui se concentre sur le partage d'informations techniques pour les entreprises utilisant Bitcoin. Bitcoin Optech offre une grande variété de ressources techniques sur Bitcoin, notamment une excellente newsletter hebdomadaire. Cette newsletter résume les discussions les plus importantes et les dernières nouvelles concernant le développement de Bitcoin et du Lightning Network. Pour vous abonner à la newsletter, rendez-vous sur : https://bitcoinops.org/en/newsletters/.

Termes associés : **BITCOINTALK** | **LIGHTNING NETWORK**

BITCOIN QT

PROTOCOLE

Bitcoin QT est un client Bitcoin intégrant une interface graphique publié en novembre 2011. Il s'inscrit dans la lignée du client de Satoshi lui-même. En 2014, Bitcoin QT est renommé « Bitcoin Core ». C'est aujourd'hui l'implémentation de référence du protocole Bitcoin. Il est fourni avec `bitcoind` depuis la version 0.5. Par ailleurs, encore aujourd'hui, l'interface graphique de Bitcoin Core s'appelle `bitcoin-qt` en référence aux origines du logiciel.

« QT » provient du nom de la bibliothèque utilisée pour l'interface graphique. Le nom « QT » est parfois interprété comme un jeu de mots sur la sonorité du terme « cute » (« mignon » en anglais).

Termes associés : **BITCOIN CORE** | **BITCOIND**

BITCOIN TREASURY COMPANY

ÉCONOMIE ET RÉGULATION | FR : SOCIÉTÉ DE TRÉSORERIE BITCOIN

Entreprise cotée en bourse qui place le bitcoin au centre de sa stratégie de trésorerie, en accumulant des bitcoins sur son bilan plutôt que de conserver ses réserves en monnaie fiat ou en obligations. L'objectif est de se protéger contre la dépréciation monétaire tout en offrant aux actionnaires une exposition indirecte au bitcoin via des actions traditionnelles.

MicroStrategy (devenue Strategy) a popularisé ce modèle sous la direction de Michael Saylor à partir d'août 2020. L'entreprise finance ses acquisitions de bitcoins par l'émission de dettes convertibles et d'actions, ce qui crée un effet de levier sur le cours du bitcoin. Plusieurs autres sociétés ont depuis adopté des stratégies similaires, comme Metaplanet au Japon ou Semler Scientific aux États-Unis.

Ce modèle permet aux investisseurs institutionnels qui ne peuvent pas détenir directement du bitcoin d'y accéder via un véhicule boursier classique, mais il expose également les actionnaires aux risques de levier financier et de volatilité du sous-jacent.

Termes associés : **ETF** | **FIAT**

BITCOIN UNLIMITED

HISTOIRE

Proposition de mise à jour du protocole Bitcoin apparue à la fin de l'été 2016, visant à augmenter la taille des blocs de manière flexible via un hard fork. Ce fork était soutenu par les big blockers durant la Blocksize War, avant notamment Roger Ver. Contrairement à Bitcoin Classic, Bitcoin Unlimited ne fixait pas de limite supérieure à la taille des blocs, mais permettait aux utilisateurs de définir leurs propres paramètres. Ces paramètres comprenaient la taille maximale de génération, la taille additionnelle des blocs et la profondeur d'acceptation. Cependant, Bitcoin Unlimited a été critiqué pour sa complexité technique et ses failles de sécurité. Cette mise à jour ne sera finalement jamais adoptée.

Terme associé : **BLOCKSIZE WAR**

BITCOIN XT

HISTOIRE

Fork de Bitcoin lancé en 2015 par le célèbre développeur Mike Hearn et soutenu par Gavin Andresen (ancien mainteneur principal de Bitcoin suite au départ de Satoshi). Bitcoin XT était à l'origine une implémentation du protocole Bitcoin compatible avec Bitcoin Core. Cependant, en août 2015, la version 0.11A de Bitcoin XT a adopté le BIP-0101 : un hard fork proposé pour augmenter la limite de la taille des blocs de 1 Mo à 8 Mo, avec une augmentation prévue pour doubler cette taille tous les deux ans jusqu'à atteindre un peu plus de 8 Go par bloc en 2036. Cette proposition a été l'un des casus belli de la Blocksize War qui a eu lieu entre 2015 et 2017. Elle était soutenue par une large partie des mineurs, et par des sociétés influentes comme BitPay, Blockchain.info ou encore Circle. Finalement, Bitcoin XT ne parviendra pas à obtenir suffisamment de soutien de la part de la communauté, et Mike Hearn finira par annoncer son départ de la communauté et la vente de ses bitcoins. Il exprimera sa déception dans un article de blog où il déclarera notamment que Bitcoin a échoué.

Terme associé : **BLOCKSIZE WAR**

BITCOIND

PROTOCOLE

Acronyme de « *Bitcoin Daemon* ». C'est un logiciel qui implémente le protocole Bitcoin et permet aux utilisateurs d'exécuter un nœud pour des appels de procédure à distance dits RPC (« *Remote Procedure Call* »). Il s'agit d'un programme en ligne de commande (sans GUI) qui sert d'interface de communication avec Bitcoin. Autrement dit, c'est un programme qui tourne en fond avec lequel l'utilisateur peut interagir (daemon). `Bitcoind` faisait partie du client original de Satoshi Nakamoto. Certains le considèrent comme le deuxième client de l'histoire de Bitcoin, après le premier de Satoshi, puisque la version 0.2.6 du logiciel permet cette exécution comme daemon sans interface graphique. Il fut par la suite regroupé avec Bitcoin QT en 2011, client renommé par la suite « Bitcoin Core », en 2014. Aujourd'hui, `bitcoind` est donc pleinement intégré au client Bitcoin Core.

Termes associés : **BITCOIN QT** **DAEMON**

BITCOIND.PID

PROTOCOLE

Fichier généré par le logiciel bitcoind (Bitcoin Daemon) lors de son exécution. Ce fichier contient l'identifiant de processus (PID) de l'instance bitcoind en cours d'exécution. Il est utilisé pour suivre et gérer le processus du logiciel, permettant à d'autres applications ou scripts de l'identifier facilement et d'interagir avec lui si nécessaire.

Termes associés : BITCOIN CORE | BITCOIND

BITCOINJ

OUTIL

Bibliothèque open source qui implémente le protocole Bitcoin en Java, créée par Mike Hearn en 2011. Elle est conçue pour fonctionner en mode SPV (*Simplified Payment Verification*) : au lieu de télécharger l'intégralité de la blockchain, elle ne récupère que les entêtes de blocs et utilise les filtres de Bloom pour identifier les transactions pertinentes. Ce fonctionnement léger la rend adaptée aux appareils mobiles et aux environnements à ressources limitées. Bitcoinj a servi de base à de nombreux projets, notamment l'app Bitcoin Wallet (qui n'existe plus), Bisq ou encore RSK.

Termes associés : IMPLÉMENTATION DE BITCOIN | SIMPLIFIED PAYMENT VERIFICATION

BITCOINTALK

COMMUNAUTÉ

Forum en ligne dédié aux discussions sur Bitcoin. Ouvert en novembre 2009 par Satoshi Nakamoto, ce forum prend la suite de l'espace de discussion dédié à Bitcoin sur sourceforge.net. Il acquiert son nom lors de la migration de bitcoin.org vers bitcointalk.org en août 2011. BitcoinTalk sert de plateforme pour l'échange d'informations, de nouvelles, de débats techniques et d'analyses.

Termes associés : BITCOIN OPTECH | BLOCKSIZE WAR

BITKEY

PORTEFEUILLE

Portefeuille matériel (*hardware wallet*) développé par Block, Inc. (anciennement Square), la société fondée par Jack Dorsey. Lancé en décembre 2023, Bitkey repose sur une architecture multisignature 2-de-3 qui répartit les clés entre le dispositif physique, une application mobile et un serveur de récupération chiffré géré par Block. Cette conception vise à simplifier la *self-custody* pour les utilisateurs non techniques en éliminant le besoin de gérer soi-même une phrase de récupération. Le dispositif se présente sous une forme compacte qui intègre un capteur d'empreinte digitale et communique avec l'application via NFC.

Termes associés : HARDWARE WALLET | MULTISIG

BITKIT

PORTEFEUILLE

Portefeuille Bitcoin mobile en *self-custody* développé par Synonym, qui combine les paiements on-chain et Lightning Network dans une même application. Bitkit propose deux comptes : « Savings » (on-chain) et « Spending » (Lightning), ainsi que le *coin control* et l'achat de cartes cadeaux via Bitrefill. Le logiciel est open source.

Termes associés : LIGHTNING NETWORK | PORTEFEUILLE

BITMAIN

MINAGE

Entreprise chinoise fondée en 2013 par Jihan Wu et Micree Zhan, spécialisée dans la conception de puces ASIC dédiées au minage de Bitcoin et la production de machines de minage. Bitmain commercialise la gamme Antminer, qui domine historiquement le marché mondial du matériel de minage en termes de parts de marché. L'entreprise a également fondé AntPool en 2014, l'une des plus grandes pools de minage, avant de s'en séparer en 2021 pour se recentrer sur la conception de puces.

Bitmain a joué un rôle central dans l'industrialisation du minage de Bitcoin en rendant les ASIC accessibles à grande échelle. L'entreprise a toutefois suscité des controverses, notamment autour de l'utilisation secrète d'AsicBoost (Covert AsicBoost) dans ses machines et de son opposition au soft fork SegWit lors du conflit sur la taille des blocs en 2017.

Termes associés : ASIC | MINING POOL

BITREFILL

OUTIL

Service en ligne sans KYC qui permet d'acheter des cartes cadeaux, des recharges téléphoniques et des abonnements en utilisant des bitcoins. Bitrefill constitue une passerelle entre l'économie Bitcoin et l'économie traditionnelle : les utilisateurs peuvent dépenser leurs bitcoins chez des centaines d'enseignes (Amazon, Uber, Steam, etc.) sans que ces dernières n'acceptent directement les paiements en bitcoins.

Le service prend en charge les paiements on-chain et via le Lightning Network, ce qui permet des transactions quasi instantanées. Bitrefill est souvent utilisé par les bitcoiners souhaitant vivre le plus possible en BTC, tout en accédant aux biens et services du commerce traditionnel.

Termes associés : LIGHTNING NETWORK | LNURL

BITVM

COUCHE SUPÉRIEURE

Protocole introduit par Robin Linus en 2023, qui vise à étendre les capacités de développement applicatif de Bitcoin. BitVM permet de réaliser n'importe quelle opération de calcul de manière arbitraire et d'utiliser ce calcul pour diriger les bitcoins engagés. Le protocole consiste à déplacer tous les calculs en dehors de la chaîne tout en permettant de contester le calcul sur la chaîne si l'autre partie prétend à un résultat frauduleux. BitVM procure ainsi à Bitcoin une capacité de calcul quasi Turing-complet, et ce, sans requérir aucune modification au niveau du consensus. BitVM reproduit le comportement d'une porte logique `NAND` grâce à une utilisation conjointe des opcodes `OP_BOOLAND` (qui reproduit lui-même le comportement d'une porte logique `AND`) et `OP_NOT` (qui reproduit le comportement d'une porte logique `NOT`). Justement, cette porte logique `NAND` peut être utilisée à la chaîne pour reproduire le comportement de toutes les autres portes logiques existantes. C'est ce que l'on appelle une « porte universelle ». Par extension, une suite de portes logiques `NAND` peut donc reproduire n'importe quel circuit de calcul. L'idée avec BitVM est de stocker ces suites de calculs `NAND` comme des feuilles dans le MAST d'une transaction Taproot.

Termes associés : **MAST** **TAPROOT**

BLIND ORACLE

COUCHE SUPÉRIEURE | FR : ORACLE AVEUGLE

Protocole cryptographique dans lequel un oracle exécute des opérations (signature, chiffrement) sur un message confidentiel transmis par un utilisateur, sans pouvoir en connaître le contenu. L'utilisateur peut ensuite vérifier la validité du résultat sans avoir exposé le message original à l'oracle. Ce mécanisme repose généralement sur des techniques de chiffrement aveugle (*blind cryptography*), comme les signatures aveugles ou les schémas d'engagement.

Dans le contexte de Bitcoin, le concept de *blind oracle* est utilisé notamment dans le *hardware wallet* Jade de Blockstream. Le portefeuille chiffre la phrase de récupération localement et utilise un oracle distant pour la protection par code PIN, sans que cet oracle n'ait accès aux clés privées ni à la phrase de récupération. Cette approche constitue un « élément sécurisé virtuel » (*virtual secure element*) qui vise à protéger contre les attaques physiques, sans nécessiter de puce sécurisée dédiée.

Termes associés : **ORACLE** **SECURE ELEMENT** **ZKP - ZERO-KNOWLEDGE PROOF**

BLINDED PATHS

LIGHTNING NETWORK | FR : CHEMINS MASQUÉS

Technique de routage Lightning qui masque une partie du chemin de paiement afin de protéger l'identité du destinataire. Dans le routage classique (SPHINX), l'expéditeur construit l'intégralité de la route et connaît donc la clé publique du destinataire. Avec les *blinded paths*, le destinataire chiffre les derniers sauts de la route grâce à un échange de clés ECDH : il dérive un secret partagé avec chaque nœud du segment masqué, génère des identifiants de nœuds aveuglés et produit des données chiffrées permettant à chaque relais de découvrir uniquement le saut suivant. L'expéditeur ne voit la route que jusqu'à un *introduction point*, sans connaître les nœuds au-delà ni le destinataire final.

Les *blinded paths* sont spécifiés dans le BOLT-04 et constituent un élément important des offres BOLT-12, où chaque facture inclut un ou plusieurs chemins masqués. Ils peuvent également être utilisés dans des factures BOLT-11 via le bLIP-0039. Les *blinded paths* permettent aussi de router des paiements à travers des canaux privés sans les révéler. Le destinataire peut ajouter des sauts factices (*dummy hops*) pour renforcer l'anonymat, car l'expéditeur ne peut pas distinguer les vrais relais des leurres.

Termes associés : **BOLT-04** **BOLT-12**

BLINK

PORTEFEUILLE

Portefeuille Bitcoin mobile custodial, anciennement connu sous le nom de « Bitcoin Beach Wallet ». Développé par Galoy, il a été créé à l'origine pour la communauté Bitcoin Beach d'El Zonte au Salvador. Il permet de gérer des soldes en BTC et en USD (via les Stablesats), et d'envoyer ou recevoir des paiements Lightning et on-chain.

Terme associé : **LIGHTNING NETWORK**

BLIP

LIGHTNING NETWORK

Acronyme de « *Bitcoin Lightning Improvement Proposal* ». Les bLIP sont des propositions de standardisation pour des fonctionnalités du Lightning Network qui ne nécessitent pas une adoption universelle, contrairement aux BOLT. Elles servent à documenter des extensions applicatives, des protocoles optionnels ou des conventions d'usage entre implémentations compatibles. Le processus de création d'un bLIP suit un cycle de révision communautaire : un auteur soumet un brouillon au dépôt GitHub dédié, qui est évalué, puis progresse du statut *Draft* à *Proposed* puis *Active* selon le consensus et l'adoption. Les bLIP complètent les BOLT en offrant un cadre plus léger et flexible pour proposer des innovations sans modifier les spécifications fondamentales du protocole.

Terme associé : **BOLT**

BLIP-0001

LIGHTNING NETWORK

Première bLIP qui établit le processus de création et de gestion des bLIP (*Bitcoin Lightning Improvement Proposals*). Elle définit le cycle de vie d'une proposition : soumission d'un brouillon, révision par les éditeurs pour la solidité technique et le formatage, puis progression à travers les statuts *Draft*, *Proposed* et *Final*/*Active*. La bLIP-0001 précise la distinction avec les BOLT : les bLIP couvrent les fonctionnalités applicatives et les extensions non universelles, tandis que les BOLT restent réservées aux changements de protocole nécessitant une adoption par toutes les implémentations.

Terme associé : **BLIP**

BLIP-0002

LIGHTNING NETWORK

bLIP qui établit un registre de réservation de valeurs pour éviter les conflits entre les différentes propositions du Lightning Network. Sans coordination, deux bLIP pourraient utiliser le même identifiant de message, le même *feature bit* ou le même type TLV, provoquant des incompatibilités réseau. La bLIP-0002 centralise les réservations dans quatre catégories : les *feature bits* (au-delà de 255), les types de messages (plage 32768-65535), les champs d'extension TLV des messages BOLT existants (`init`, `payment_onion_payload`, `update_add_htlc`, `ping`) et les charges utiles de messages en oignon. Chaque entrée est liée au bLIP qui l'utilise.

Terme associé : **BLIP**

BLIP-0003

LIGHTNING NETWORK

bLIP qui formalise la spécification du *keysend*, un mécanisme de paiement spontané sur le Lightning Network. Avec le *keysend*, l'expéditeur génère une préimage aléatoire de 32 octets, l'inclut dans un enregistrement TLV (type `5482373484`) du paquet en oignon et utilise son hash SHA-256 comme hash de paiement. Le destinataire vérifie la correspondance entre la préimage et le hash pour valider le paiement. Le *keysend* devrait à terme être remplacé par des mécanismes plus complets comme les offres BOLT-12.

Terme associé : **BLIP**

BLIP-0004

LIGHTNING NETWORK

bLIP qui propose un mécanisme expérimental de signalement d'endossement (*endorsement*) des HTLC pour lutter contre les attaques de *channel jamming* sur le Lightning Network. Chaque HTLC transporte un champ TLV d'un octet indiquant s'il est endossé (valeur 7) ou non (valeur 0). Les nœuds d'origine marquent comme endossés les paiements dont ils anticipent le succès. Les nœuds intermédiaires propagent le signal positif s'ils sont compatibles, sinon transmettent la valeur zéro par défaut. Ce mécanisme est purement observationnel : les données collectées servent à tester des algorithmes de réputation sans affecter le routage réel des paiements. L'expérience est limitée dans le temps par une date d'expiration, après laquelle les nœuds cessent de relayer le signal.

Terme associé : **BLIP**

BLIP-0010

LIGHTNING NETWORK

bLIP qui standardise le format de métadonnées attaché aux paiements Lightning dans les applications de podcasting (*Podcasting 2.0*). Elle définit une structure JSON transmise via un enregistrement TLV (type 7629169) accompagnant chaque paiement. Ces métadonnées identifient le podcast, l'épisode, l'horodatage de lecture, le montant en millisatoshis, les informations de l'expéditeur et un message optionnel.

Le système supporte deux types de paiements : les *streams* (micropaiements automatiques à la minute d'écoute) et les *boosts* (contributions manuelles pouvant inclure un message appelé « *boostagram* »). Cette standardisation permet à n'importe quel portefeuille compatible d'envoyer des paiements que tout hébergeur de podcast compatible peut traiter de manière uniforme.

Terme associé : **BLIP**

BLIP-0011

LIGHTNING NETWORK

bLIP qui définit *NameDesc*, un format standardisé pour structurer les descriptions des factures BOLT-11 en y intégrant le nom du destinataire. La convention consiste à préfixer la description par le nom du commerçant ou du destinataire, suivi d'un deux-points et de deux espaces (`: `), puis de la description du paiement.

Les portefeuilles compatibles peuvent ainsi extraire et afficher séparément le nom du destinataire. Les portefeuilles non compatibles affichent simplement le texte complet comme une description classique, ce qui garantit une rétrocompatibilité totale. Cette convention résout le problème des historiques de transactions où seul un identifiant technique ou un nom de produit apparaît, sans identification claire du commerçant.

Termes associés : **BLIP** **BOLT-11**

BLIP-0017

LIGHTNING NETWORK

bLIP qui spécifie les *hosted channels*, des canaux de paiement Lightning reposant sur la confiance du client envers un hôte, sans ancrage on-chain. Contrairement aux canaux Lightning classiques qui verrouillent des bitcoins dans une transaction de financement, les canaux hébergés fonctionnent sans UTXO ni interaction avec la blockchain. L'état du canal est signé par les deux parties via des messages chiffrés échangés sur le protocole Lightning, ce qui empêche l'hôte de falsifier les soldes unilatéralement. Les avantages incluent une capacité de réception immédiate et l'absence de frais on-chain, mais le client assume un risque de contrepartie car il ne peut pas forcer la fermeture du canal sur la blockchain. Ce modèle se positionne entre un portefeuille custodial classique et un canal Lightning classique.

Terme associé : **BLIP**

BLIP-0025

LIGHTNING NETWORK

bLIP qui autorise l'avant-dernier saut d'une route Lightning à transmettre un HTLC d'un montant inférieur à celui encodé dans le paquet en oignon. Le nœud relais déclare les frais supplémentaires qu'il a prélevés via un nouveau champ TLV appelé `extra_fee`. Ce mécanisme est principalement conçu pour les fournisseurs de services Lightning (LSP) qui ouvrent des canaux *just-in-time* pour de nouveaux utilisateurs. En prélevant des frais additionnels sur le premier paiement reçu, le LSP peut compenser le coût on-chain d'ouverture du canal. Le destinataire vérifie explicitement le montant des frais prélevés, ce qui empêche tout abus de la part du nœud intermédiaire.

Terme associé : **BLIP**

BLIP-0032

LIGHTNING NETWORK

bLIP qui définit un protocole de résolution DNS privée via les messages en oignon du Lightning Network. Elle permet à un nœud de résoudre des noms de domaine en enregistrements TXT avec preuves DNSSEC, sans passer par un serveur DNS public. Le protocole introduit deux types de messages en oignon : `dnssec_query` (requête de résolution) et `dnssec_proof` (réponse avec preuve cryptographique).

Le cas d'usage principal est la résolution d'identifiants lisibles au format `utilisateur@domaine` en instructions de paiement, en s'intégrant avec les offres BOLT-12. Le destinataire publie ses instructions de paiement dans un enregistrement DNS TXT, et le payeur les récupère via le réseau Lightning sans exposer sa requête à l'infrastructure DNS publique, préservant ainsi sa vie privée.

Termes associés : **BLIP** **BOLT-12**

BLIP-0039

LIGHTNING NETWORK

bLIP qui introduit un nouveau champ étiqueté `b` dans les factures BOLT-11 pour transmettre des chemins de paiement masqués (*blinded paths*). Cette extension permet d'utiliser le *route blinding* (initialement conçu pour BOLT-12) directement dans les factures BOLT-11 existantes, améliorant la confidentialité du destinataire avant le déploiement complet des offres BOLT-12.

Termes associés : **BLIP** **BOLT-11**

BLIP-0050

LIGHTNING NETWORK

bLIP qui définit LSPS0, la couche de transport standardisée pour la communication entre les fournisseurs de services Lightning (LSP) et leurs clients. Le protocole réutilise les connexions pair-à-pair BOLT-08 existantes et y transporte des messages JSON-RPC 2.0 via un identifiant de message unique (`37913`). Le LSP agit comme serveur JSON-RPC et le client comme client, avec des paramètres nommés et des réponses potentiellement désordonnées. LSPS0 définit également les schémas communs d'encodage pour les montants, les frais, les signatures et les adresses utilisés par toutes les spécifications LSPS de niveau supérieur (LSPS1 à LSPS5). Ce choix architectural évite d'introduire des dépendances externes comme HTTP ou gRPC.

Terme associé : **BLIP**

BLIP-0051

LIGHTNING NETWORK

bLIP qui définit LSPS1, une API standardisée permettant aux portefeuilles Lightning d'acheter des canaux de paiement auprès de fournisseurs de services Lightning (LSP). Le processus se déroule en trois étapes :

- le client interroge les options disponibles du LSP via `lsps1.get_info`,
- il crée une commande avec les paramètres souhaités (capacité, durée, répartition des soldes) via `lsps1.create_order`,
- puis il règle via une facture BOLT-11, une offre BOLT-12 ou un paiement on-chain.

Le LSP ouvre ensuite le canal demandé. La spécification supporte les canaux publics et privés, les canaux à zéro confirmation, et prévoit un remboursement automatique en cas d'échec de l'ouverture. LSPS1 repose sur la couche de transport LSPS0 du bLIP-0050.

Termes associés : **BLIP** **BLIP-0050**

BLIP-0052

LIGHTNING NETWORK

bLIP qui définit LSPS2, un protocole de négociation de canaux *just-in-time* (JIT) sur le Lightning Network. Il permet à un client sans canal existant de recevoir un paiement entrant : le LSP ouvre un canal à la volée lorsqu'un paiement arrive, et déduit les frais d'ouverture directement du montant transféré.

Termes associés : **BLIP** **BLIP-0050**

BLIP-0055

LIGHTNING NETWORK

bLIP qui définit LSPS5, un protocole d'enregistrement de *webhooks* permettant aux fournisseurs de services Lightning (LSP) de notifier les clients mobiles lorsqu'un événement nécessite leur attention. Les applications mobiles, suspendues par le système d'exploitation lorsqu'elles ne sont pas actives, perdent leurs connexions réseau et ne peuvent pas recevoir de paiements ou traiter des HTLC expirants. Avec LSPS5, le client enregistre une URL de *webhook* auprès du LSP via trois API : `lsps5.set_webhook`, `lsps5.list_webhooks` et `lsps5.remove_webhook`. Le LSP contacte ce *webhook* lorsqu'un paiement entrant arrive, qu'un HTLC est sur le point d'expirer, que la liquidité change ou qu'un message en oignon doit être livré. Les notifications sont signées cryptographiquement par la clé d'identité du nœud LSP.

Termes associés : **BLIP** **BLIP-0050**

BLIXT WALLET

PORTEFEUILLE

Portefeuille Lightning *open source* et *non-custodial* pour Android et iOS. Blixt embarque un nœud LND directement sur le téléphone de l'utilisateur, grâce à Neutrino.

Termes associés : **LIGHTNING NETWORK** **NOEUD LIGHTNING**

BLK*.DAT

PROTOCOLE

Nom des fichiers dans Bitcoin Core qui stockent les données brutes des blocs de la blockchain. Chaque fichier est identifié par un numéro unique dans son nom. Ainsi, les blocs sont enregistrés dans l'ordre chronologique, en commençant par le fichier blk00000.dat. Lorsque ce fichier atteint sa capacité maximale, les blocs suivants sont enregistrés dans blk00001.dat, puis blk00002.dat, et ainsi de suite. Chaque fichier blk a une capacité maximale de 128 mébioctets (MiB), ce qui équivaut à un peu plus de 134 mégaoctets (Mo). Ces fichiers ont été déplacés dans le dossier `/blocks` depuis la version 0.8.0.

Termes associés : **BLOCKS INDEX** **BLOCKS/REV*.DAT**

BLKINDEX.DAT

PROTOCOLE

Nom de l'ancien fichier utilisé dans Bitcoin Core pour stocker diverses informations sur la blockchain, remplacé depuis la version 0.8.0 par les fichiers dans chainstate/, blocks/index/ et blocks/rev*.dat.

Termes associés : **BLOCKS INDEX** | **CHAINSTATE/**

BLKTREE/

PROTOCOLE

Nom de l'ancien dossier utilisé dans Bitcoin Core pour cataloguer les métadonnées sur tous les blocs. Ce dossier a été remplacé par le dossier `blocks/index/` dans la version `0.8.0`.

Termes associés : **BLK*.DAT** | **BLOCKS INDEX**

BLOC

PROTOCOLE | EN : BLOCK

Structure de données dans le système Bitcoin. Un bloc contient un ensemble de transactions valides et des métadonnées contenues dans son entête. Chaque bloc est lié au suivant par le hachage de son entête, formant ainsi la blockchain (chaîne de blocs). La blockchain agit comme un serveur d'horodatage qui permet à chaque utilisateur de connaître l'ensemble des transactions passées, afin de vérifier la non-existence d'une transaction et éviter la double dépense. Les transactions sont organisées dans un arbre de Merkle. Cet accumulateur cryptographique permet de produire un condensat de toutes les transactions d'un bloc, appelé « Racine de Merkle » (*Merkle root*). L'entête d'un bloc contient 6 éléments :

- La version du bloc ;
- L'empreinte du bloc précédent ;
- La racine de l'arbre de Merkle des transactions ;
- L'horodatage du bloc ;
- La cible de difficulté ;
- Le nonce.

Pour être valide, un bloc doit disposer d'un entête qui, une fois haché avec `SHA256d`, produit un condensat inférieur ou égal à la cible de difficulté.

Termes associés : **ARBRE DE MERKLE** | **BLOCKCHAIN**

BLOCK-RELAY-ONLY

RÉSEAU

Type de connexion sortante dans Bitcoin Core qui ne relaie que les blocs, sans échanger de transactions ni d'adresses de nœuds. Depuis la version 0.19.0 (novembre 2019), chaque nœud établit par défaut 2 connexions *block-relay-only* en plus de ses 8 connexions sortantes classiques.

Ces connexions renforcent la résistance au partitionnement du réseau et la protection contre les attaques éclipse, car elles augmentent la connectivité du nœud sans révéler d'information sur la topologie du réseau. En effet, les transactions et les adresses constituent les principaux vecteurs par lesquels un adversaire peut cartographier les liens entre nœuds. L'absence de ces échanges rend les connexions *block-relay-only* invisibles pour un espion qui analyse la propagation des transactions. Elles garantissent ainsi au nœud un accès à la blockchain légitime même si toutes ses connexions classiques sont compromises.

Termes associés : **ECLIPSE** | **PAIR SORTANT**

BLOCK TEMPLATE

MINAGE | FR : MODÈLE DE BLOC

Ensemble d'informations fournies par un nœud Bitcoin à un logiciel de minage ou une pool, contenant les données nécessaires pour construire un nouveau bloc candidat. Cette structure inclut les transactions sélectionnées pour être incluses et l'entête du bloc. Une fois que le mineur reçoit ces informations, il est en capacité de commencer à chercher une preuve de travail valide pour le bloc candidat.

Termes associés : **CANDIDAT - BLOC** | **STRATUM V2**

BLOCKCHAIN

PROTOCOLE | FR : CHAÎNE DE BLOCS

La blockchain est le nom communément donné au serveur d'horodatage distribué du système Bitcoin. C'est une chaîne de blocs. Chaque bloc est lié au suivant par son empreinte cryptographique. Pour éviter la double dépense sur Bitcoin sans recourir à une autorité centrale, il faut que chaque utilisateur vérifie la non-existence d'une transaction dans le système. Le seul moyen de s'assurer de l'absence d'une transaction est d'être au courant de toutes les transactions Bitcoin passées. Dans cet objectif, les transactions sont horodatées au sein de blocs, et chaque utilisateur dispose de l'entièreté de la blockchain.

Suite aux nombreuses utilisations marketing abusives du terme de « Blockchain », notamment à la fin des années 2010, beaucoup de bitcoiners refusent l'emploi de ce mot. Certains préfèrent parler de « TimeChain » pour évoquer ce concept. D'autres, se référant au White Paper de Satoshi Nakamoto, évoquent une « Proof-of-Work Chain ». En français, le terme anglais de « Blockchain » est globalement admis. On peut également utiliser la traduction « chaîne de blocs ».

Termes associés : **BLOC** | **PREUVE DE TRAVAIL**

BLOCKS INDEX

PROTOCOLE

Structure de données LevelDB dans Bitcoin Core qui catalogue des métadonnées sur tous les blocs. Chaque entrée dans cet index renseigne des détails tels que l'identifiant du bloc, sa hauteur dans la blockchain, le pointeur vers le bloc dans la base de données, et d'autres métadonnées. Cette indexation permet de trouver rapidement un bloc stocké en mémoire.

Termes associés : **BLK*.DAT** | **LEVELDB**

BLOCKS/REV*.DAT

PROTOCOLE

Nom des fichiers dans Bitcoin Core qui stockent les informations nécessaires pour éventuellement annuler les modifications apportées à l'UTXO set par les blocs précédemment ajoutés. Chaque fichier est identifié par un numéro unique qui est le même que le fichier blk.*dat auquel il correspond. Les fichiers rev*.dat contiennent les données d'annulation correspondant aux blocs stockés dans les fichiers blk*.dat. Ces données sont essentiellement une liste de tous les UTXO dépensés en input dans un bloc. Ces fichiers d'annulation permettent au nœud de revenir à un état antérieur en cas de réorganisation de la blockchain provoquant l'abandon de blocs préalablement valides.

Termes associés : **BLK*.DAT** | **UTXO SET**

BLOCKSIGNERS

SIDECHAIN | FR : SIGNATAIRES DE BLOCS

Dans le contexte de Liquid (sidechain de Bitcoin), ce sont les entités responsables de la construction et de la validation des blocs au sein de cette chaîne. Liquid utilise un modèle de fédération où les *blocksigners*, sélectionnés parmi les membres de la fédération, opèrent conjointement pour confirmer les transactions et créer de nouveaux blocs afin de former le consensus de la sidechain. Le rôle de *blocksigner* fait partie des fonctionnaires dans Liquid. Ces derniers assurent à la fois ce rôle, et celui de *watchmen* (gardiens).

Termes associés : **FONCTIONNAIRE** | **LIQUID NETWORK**

BLOCKSIZE WAR

HISTOIRE | FR : GUERRE DE LA TAILLE DES BLOCS

Désigne une période de débat intense et de conflit au sein de la communauté Bitcoin entre 2015 et 2017. Ce conflit portait sur la question de savoir si la taille des blocs, limitée à 1 mégaoctet depuis 2010 par Satoshi Nakamoto, devait être augmentée pour permettre à Bitcoin de traiter plus de transactions par bloc, et ainsi passer à l'échelle (scalabilité).

Les partisans de l'augmentation de la taille des blocs (que l'on appelle les « *big blockers* »), menés par des développeurs de la première heure comme Jeff Garzik, Gavin Andresen et Mike Hearn, des figures influentes de l'écosystème comme Roger Ver, ainsi qu'une bonne part des grandes entreprises de minage, soutenaient que cette augmentation était nécessaire pour réduire les frais de transaction et améliorer la scalabilité de Bitcoin. Ils pensaient que cette mo-

dification pourrait se faire sans engendrer de conséquences sur la sécurité du système. D'autre part, les opposants à l'augmentation de la taille des blocs (que l'on appelle les « *small blockers* »), principalement des développeurs de Bitcoin Core (Pieter Wuille, Peter Todd, Gregory Maxwell, Wladimir van der Laan, Luke Dashjr...) et une partie de la communauté technique, avançaient que cette modification centraliserait le système à cause de l'augmentation de la charge opérationnelle sur les nœuds. Ils mettaient en avant des solutions d'évolutivité dites de seconde couche, comme le Lightning Network, qui permettraient de prendre en charge une grande quantité de transactions, sans pour autant devoir modifier le protocole de base.

Le débat sur l'approche de Bitcoin en termes de scalabilité dure au moins depuis 2010, avec la première tentative d'augmentation de la taille des blocs en octobre de cette année via un patch proposé par Jeff Garzik. Mais c'est bien durant la période de la Blocksize War entre 2015 et 2017 que les conflits ont été les plus virulents avec la mise en place de plusieurs tentatives de forks (Bitcoin XT, Bitcoin Classic, Bitcoin Unlimited, SegWit, SegWit2X et Bitcoin Cash), avec des stratégies de propagande et d'influence, ainsi qu'avec des attaques personnelles. L'abandon du hard fork SegWit2X le 8 novembre 2017 faisant suite à l'opposition massive de la communauté (« NO2X ») est souvent vu comme l'évènement marquant la fin de la Blocksize War, tandis que le casus belli est parfois considéré comme étant la mise à jour 0.11A du client Bitcoin XT, qui a implémenté le BIP-0101 le 15 août 2015. On pourrait également estimer que la fin de la Blocksize War a eu lieu le 24 août 2017, date à laquelle le soft fork SegWit a été activé suite à la campagne en faveur de l'« UASF ».

En français, on parle généralement de la « Blocksize War » comme un nom propre. Certains traduisent parfois ce terme par « guerre des blocs » ou « guerre de la taille des blocs ».

Termes associés : **SEGWIT** | **UASF**

BLOCKSTREAM

ORGANISATION

Entreprise spécialisée dans le développement de solutions autour de Bitcoin. Blockstream est à l'initiative de la sidechain Liquid, de l'implémentation du Lightning Network Core Lightning ou encore des portefeuilles Jade et Green. Elle est également connue pour employer des développeurs Bitcoin Core. La société Blockstream est actuellement dirigée par le cypherpunk et cryptographe Adam Back, l'inventeur de Hashcash, le protocole qui a inspiré la preuve de travail sur Bitcoin.

Termes associés : **HASHCASH** | **LIQUID NETWORK**

BLOOM FILTER

RÉSEAU | FR : FILTRE DE BLOOM

Structure de données probabiliste utilisée pour tester si un élément fait partie d'un ensemble. Les Bloom Filters permettent de vérifier rapidement l'appartenance sans tester l'ensemble des données. Ils sont particulièrement utiles dans les contextes où l'espace et la vitesse sont critiques, mais où un taux d'erreur faible et contrôlé est acceptable. En effet, les Bloom Filters ne produisent pas de faux négatifs, mais ils produisent une certaine quantité de faux positifs. Lorsqu'un élément est ajouté au filtre, plusieurs fonctions de hachage génèrent des positions dans un tableau de bits. Pour vérifier l'appartenance, les mêmes fonctions de hachage sont utilisées. Si tous les bits correspondants sont définis, l'élément est probablement dans l'ensemble, mais avec un risque de faux positifs. Les filtres de Bloom sont largement utilisés dans le domaine des bases de données et des réseaux. On sait notamment que Google les utilise pour son système de stockage distribué pour données structurées *BigTable*. Dans le protocole Bitcoin, on les utilise notamment pour les portefeuilles SPV selon le BIP-0037.

Lorsque l'on parle spécifiquement de l'utilisation des Bloom Filters dans le cadre des transactions Bitcoin, on retrouve parfois le terme de « Transaction Bloom Filtering ».

Terme associé : **BIP-0037**

BLUE WALLET

PORTEFEUILLE

Portefeuille Bitcoin mobile *open source* disponible sur iOS et Android. L'application permet de créer plusieurs portefeuilles au sein de la même interface, de gérer les PSBT pour l'utilisation avec des *hardware wallets* et d'importer des portefeuilles en *watch-only*. BlueWallet peut être connecté à son propre noeud Bitcoin et à son propre serveur Electrum pour une utilisation souveraine. Il offre également des fonctionnalités avancées comme le *coin control*, le remplacement de transaction par RBF et la personnalisation des frais.

Termes associés : **LIGHTNING NETWORK** | **PORTEFEUILLE**

BOLT

LIGHTNING NETWORK

Acronyme de « *Basis of Lightning Technology* ». C'est une série de spécifications destinées à permettre l'interopérabilité de Lightning entre les différentes implémentations de ce protocole de seconde couche (*LND*, *Core Lightning*, *Eclair*...). Ces spécifications détaillent les règles et les normes à respecter afin que les nœuds Lightning forment un seul et même réseau.

Termes associés : **ECLAIR** | **LND**

BOLT-00

LIGHTNING NETWORK

Document d'introduction et d'index des spécifications BOLT (*Basis of Lightning Technology*). Il présente une vue d'ensemble du fonctionnement du Lightning Network, ainsi que chaque BOLT.

Terme associé : **BOLT**

BOLT-01

LIGHTNING NETWORK

Spécification qui définit le protocole de base de la messagerie entre nœuds Lightning. Elle établit le format des messages, leur classification en cinq catégories (configuration, canal, engagement, routage et messages personnalisés) et les types de données fondamentaux. Le BOLT-01 introduit le format TLV (*Type-Length-Value*), un encodage extensible qui permet aux implémentations d'ignorer les champs inconnus tout en maintenant la compatibilité. Il spécifie également le message `init` échangé à la connexion entre deux pairs pour déclarer leurs fonctionnalités, les messages `error` et `warning` pour la gestion des erreurs, et le mécanisme `ping`/`pong` pour maintenir les connexions actives.

Terme associé : **BOLT**

BOLT-02

LIGHTNING NETWORK

Spécification qui définit le protocole pair-à-pair de gestion des canaux Lightning. Elle couvre l'intégralité du cycle de vie d'un canal : l'établissement (en version classique avec un seul financeur, et en version 2 avec financement bilatéral via construction interactive de transaction), le fonctionnement normal (ajout et résolution de HTLC, mise à jour des transactions d'engagement) et la fermeture coopérative. Le BOLT-02 spécifie également les mécanismes d'augmentation des frais par RBF, la mise au repos (*quiescence*) du canal pour les mises à jour de protocole, et le protocole de retransmission de messages en cas de reconnexion entre pairs.

Termes associés : **BOLT** **TRANSACTION DE FINANCEMENT**

BOLT-03

LIGHTNING NETWORK

Spécification qui définit les formats exacts des transactions Bitcoin et des scripts utilisés dans les canaux Lightning. Elle détaille la structure de la transaction de financement (sortie P2WSH multisignature 2/2), des transactions d'engagement (*commitment transactions*) qui représentent l'état courant du canal avec leurs sorties `to_local` et `to_remote`, et des transactions HTLC. Le BOLT-03 spécifie également le calcul des frais, les limites de poussière (*dust*), les procédures de dérivation des clés et les poids attendus des transactions pour une estimation correcte des frais.

Termes associés : **BOLT** **TRANSACTION D'ENGAGEMENT**

BOLT-04

LIGHTNING NETWORK

Spécification qui définit le protocole de routage en oignon du Lightning Network. Elle décrit la construction et le traitement de paquets chiffrés de 1 366 octets contenant une clé éphémère, des instructions de routage imbriquées et un HMAC d'intégrité. Chaque nœud intermédiaire ne voit que le saut suivant, sans connaître l'expéditeur ni le destinataire final.

Le BOLT-04 couvre aussi le *route blinding*, qui masque l'identité des nœuds en aval pour protéger la vie privée du destinataire, et le mécanisme de retour d'erreurs chiffrées qui permet à l'expéditeur d'identifier quel nœud a échoué.

Termes associés : BOLT | ROUTAGE EN OIGNON

BOLT-05

LIGHTNING NETWORK

Spécification qui définit le comportement des nœuds Lightning lorsqu'un canal est fermé on-chain. Elle couvre trois scénarios : la fermeture coopérative, la fermeture unilatérale (publication d'une transaction d'engagement) et la fermeture par révocation (tentative de fraude). Le BOLT-05 détaille la résolution de chaque type de sortie (sorties locales, distantes, d'ancrage et HTLC), les règles de surveillance de la blockchain jusqu'à ce que tous les fonds soient irrévocablement résolus (100 blocs de profondeur), et les transactions de pénalité permettant de saisir la totalité des fonds du canal lorsqu'un ancien état révoqué est publié. La spécification précise également les mécanismes d'augmentation de frais (*fee bumping*) via les sorties d'ancrage.

Terme associé : BOLT

BOLT-07

LIGHTNING NETWORK

Spécification qui définit le protocole de découverte pair-à-pair du réseau Lightning. Elle décrit les messages `node_announcement` (par lesquels les nœuds diffusent leur identité et leurs adresses de connexion), `channel_announcement` (qui prouvent l'existence d'un canal) et `channel_update` (qui publient les frais de routage et les délais d'expiration de chaque canal).

Le BOLT-07 établit les règles de diffusion échelonnée des annonces (*gossip*), le mécanisme d'élagage des canaux obsolètes ou fermés on-chain, et les protocoles de synchronisation par requêtes permettant aux nœuds de maintenir une vue cohérente de la topologie du réseau.

Terme associé : BOLT

BOLT-08

LIGHTNING NETWORK

Spécification qui définit la couche de transport chiffrée et authentifiée du Lightning Network. Elle repose sur le protocole Noise, où l'initiateur doit connaître la clé publique du répondeur avant la connexion. Le *handshake* se déroule en trois actes, au terme desquels les deux pairs partagent des clés de session.

Terme associé : **BOLT**

BOLT-09

LIGHTNING NETWORK

Spécification qui établit le registre centralisé des *feature bits* utilisés dans le Lightning Network pour signaler les fonctionnalités supportées par chaque nœud. Les fonctionnalités sont annoncées par des *flags* dans les messages `init`, `node_announcement` et `channel_announcement`. Chaque fonctionnalité utilise une paire de bits : le bit pair indique un support obligatoire, le bit impair indique un support optionnel.

Terme associé : **BOLT**

BOLT-10

LIGHTNING NETWORK

Spécification qui définit un mécanisme de découverte de nœuds Lightning basé sur le DNS. Elle permet à un nœud qui démarre sans contacts connus de trouver des pairs initiaux (*bootstrap*) en interrogeant des serveurs DNS autoritaires (*seed nodes*). Les requêtes supportent des filtres par type d'adresse (IPv4, IPv6), par identifiant de nœud spécifique et par nombre de résultats souhaités. Les serveurs DNS renvoient un échantillon aléatoire de nœuds sains, avec les adresses IP via des enregistrements `A`/`AAAA` (port standard `9735`) et des enregistrements `SRV` pour les ports non standard.

Terme associé : **BOLT**

BOLT-11

LIGHTNING NETWORK

Spécification qui définit le format d'encodage des factures de paiement Lightning (*payment requests*), conçu pour être compatible avec les codes QR. Une facture BOLT-11 est encodée en Bech32 et se compose de trois parties : un préfixe lisible indiquant le réseau et un montant optionnel (avec des multiplicateurs `m`, `u`, `n`, `p`), une section de données contenant un horodatage et des champs étiquetés (*tagged fields*), et une signature ECDSA. Les champs étiquetés transportent le hash de paiement, le secret de paiement, la description, le délai d'expiration, des indices de routage pour les canaux privés et des flags de fonctionnalités. BOLT-11 est progressivement complété par BOLT-12 qui apporte des factures réutilisables et une meilleure confidentialité.

Termes associés : **BOLT** **BOLT-12**

BOLT-12

LIGHTNING NETWORK

Spécification qui définit un protocole de négociation de paiement Lightning plus flexible que BOLT-11. Elle introduit trois concepts : les *offers* (modèles de paiement réutilisables publiables via QR codes), les *invoice requests* (demandes d'invoice envoyées via le réseau Lightning en réponse à une offre), et les invoices elles-mêmes.

Contrairement aux invoices BOLT-11 à usage unique, les *offers* BOLT-12 peuvent être réutilisées indéfiniment et restent sans état côté commerçant. Le format remplace les *tagged fields* de BOLT-11 par un encodage TLV (*Type-Length-Value*) plus extensible, tout en conservant un encodage bech32-style, supporte la divulgation sélective de champs, permet les paiements inter-devises et améliore la confidentialité via les *blinded paths* qui masquent l'identité du destinataire.

Termes associés : **BOLT** | **BOLT-11**

BOLTZ

OUTIL

Service *open source* permettant de réaliser des *submarine swaps* et des *atomic swaps* de manière *non-custodial* entre Bitcoin on-chain, le Lightning Network et le réseau Liquid. Lancé en 2019, Boltz utilise des HTLC pour garantir que les échanges se déroulent de façon *trustless*, sans qu'aucune des parties ne doive faire confiance à l'autre. Le service est parfois intégré directement dans des applications de portefeuille, afin de permettre d'utiliser Lightning sans avoir de nœud, mais tout en restant en *self-custody*.

Termes associés : **ATOMIC SWAP** | **SUBMARINE SWAP**

BORROMEAN RING SIGNATURES

CRYPTOGRAPHIE | FR : SIGNATURES EN ANNEAU DE BORROMÉE

Protocole cryptographique de signatures en anneau qui permet à un signataire de prouver qu'il détient une clé privée parmi un ensemble de clés publiques, sans révéler laquelle. Le nom fait référence aux anneaux borroméens, une structure topologique dans laquelle trois anneaux sont liés de telle sorte que la suppression de n'importe lequel libère les deux autres. Ce schéma a été formalisé par Gregory Maxwell et Andrew Poelstra en 2015.

Dans le contexte de Bitcoin, les *Borromean ring signatures* ont été utilisées sur le réseau Liquid comme mécanisme initial de *rangeproofs* au sein des *Confidential Transactions*. Elles permettaient de prouver qu'un montant engagé se situait dans un intervalle valide, sans en révéler la valeur. Ce rôle a par la suite été repris par les *Bulletproofs*, qui offrent des preuves plus compactes.

Termes associés : **BULLETPROOFS** | **CONFIDENTIAL TRANSACTIONS** | **RANGEPROOFS**

BOS - BALANCE OF SATOSHIS

OUTIL

Outil en ligne de commande développé par Alex Bosworth pour gérer et optimiser un nœud Lightning fonctionnant sous LND. Il s'installe via npm (`npm install -g balanceofsatoshis`) et offre un ensemble de commandes accessibles par le préfixe `bos` pour consulter les balances, ajuster les frais de routage, rééquilibrer la liquidité entre canaux, ouvrir ou fermer des canaux, et surveiller l'activité du nœud.

Parmi ses fonctionnalités principales, `bos rebalance` permet de déplacer de la liquidité entre canaux via des paiements circulaires, `bos fees` d'ajuster dynamiquement les frais sortants avec des formules conditionnelles, et `bos accounting` de produire un relevé comptable des transactions. L'outil propose également une intégration Telegram pour recevoir des notifications et interagir avec son nœud à distance.

Termes associés : **LND** **UMBREL**

BOSMINER

MINAGE

Logiciel de minage *open-source* développé en Rust par Braiins en tant qu'alternative plus moderne à CGMiner. BOSminer est spécifiquement conçu pour les ASICs. Intégré à l'écosystème Braiins OS, il permet une gestion avancée des ASICs, avec des fonctionnalités comme l'overclocking, une surveillance détaillée des performances et le support de Stratum V2.

Termes associés : **CGMINER** **STRATUM V2**

BOUNTY

COMMUNAUTÉ

Récompense offerte pour la réalisation d'une tâche spécifique, souvent liée à l'amélioration ou à la sécurité d'un projet de logiciel. Les bounties sont souvent utilisés pour encourager la découverte et la correction de vulnérabilités dans les logiciels, pour développer de nouvelles fonctionnalités, ou pour résoudre des problèmes complexes. Les participants, généralement des développeurs ou des chercheurs, reçoivent une récompense une fois que leurs contributions ont été vérifiées et acceptées. Ce système incitatif permet d'accélérer le développement et de renforcer la sécurité des logiciels en mobilisant une communauté plus large. Les bounties sont parfois utilisés par des logiciels dans le cadre de Bitcoin.

Termes associés : **FOSS** **GITHUB**

BPM - BITCOIN POOLED MINING

↗ MINAGE

Autre nom donné à la méthode « SCORE BASED METHOD » pour le calcul de la rémunération des mineurs dans le contexte des pools de minage. C'est une méthode de récompense proportionnelle, mais pondérée par le moment auquel la share est soumise. Le BPM valorise les shares en fonction du temps écoulé depuis le début du cycle de minage. Plus une share est soumise tardivement dans le cycle, plus sa valeur est élevée. Cette méthode permet d'inciter les mineurs à rester, car à chaque arrêt du minage, le mineur voit son score stagner alors que celui des autres augmente de plus en plus rapidement.

« *Bitcoin Pooled Mining* » était également le nom de la pool de minage de Slush en 2010, qui est à l'origine de la méthode de calcul des rémunérations de même nom.

Terme associé : **SHARES**

BRAIDPOOL

↗ MINAGE

Proposition de pool de minage décentralisée pour Bitcoin, conçue par Bob McElrath. Braidpool vise à résoudre le problème de centralisation des pools de minage traditionnelles, où l'opérateur de la pool sélectionne les transactions et contrôle la construction des blocs.

Le protocole repose sur un DAG (*Directed Acyclic Graph*) miné en parallèle de la blockchain Bitcoin. Les blocs du DAG sont des « blocs faibles » : des blocs Bitcoin valides qui ne satisfont pas la cible de difficulté du réseau principal, mais seulement une cible réduite propre à Braidpool. Ces blocs faibles jouent le rôle de shares et permettent de mesurer la contribution de chaque mineur. Contrairement à P2Pool qui utilise une sharechain linéaire, la structure en DAG tolère mieux la latence réseau et réduit le taux de blocs obsolètes (orphelins).

Chaque mineur construit ses propres modèles de blocs et choisit les transactions à inclure, ce qui préserve la décentralisation du consensus Bitcoin. À la fin de chaque période d'ajustement de la difficulté, les récompenses sont distribuées aux mineurs via une transaction multisig qui utilise le protocole FROST. Braidpool est encore en cours de développement et n'est pas encore déployé en production.

Termes associés : **MINING POOL** **P2POOL**

BRAIINS POOL

↗ MINAGE

Pool de minage de Bitcoin exploitée par l'entreprise tchèque Braiins. Créée en 2010 par Marek Palatinus (« Slush »), c'est la toute première pool de minage de Bitcoin, et elle a miné son premier bloc le 16 décembre 2010. Initialement connue sous le nom de « Bitcoin.cz Mining Pool », puis « Slush Pool » à partir de 2016, elle a été renommée « Braiins Pool » en 2022. Braiins est également à l'origine du protocole Stratum V2, une mise à jour majeure du protocole de communication entre les mineurs et les pools, qui améliore la sécurité, réduit la bande passante et permet aux mineurs de sélectionner eux-mêmes les transactions à inclure dans les blocs. L'entreprise développe aussi Braiins OS, un firmware open source pour les machines ASIC.

Termes associés : **MINING POOL** | **STRATUM V2**

BRAINWALLET

⊸ PORTEFEUILLE | FR : PORTEFEUILLE CERVEAU

Méthode de sauvegarde d'un portefeuille Bitcoin qui consiste à ne conserver sa phrase mnémonique sur aucun support physique ou numérique, et à la retenir uniquement de tête. Par extension, le terme désigne parfois aussi le fait de choisir soi-même les mots de sa phrase mnémonique (au lieu de la générer aléatoirement) afin de la mémoriser plus facilement. Dans les deux cas, l'objectif est de ne dépendre d'aucun support matériel.

Cette approche est doublement dangereuse. Premièrement, la mémoire humaine est faillible : un oubli, un accident ou un décès entraîne une perte définitive des fonds, et pose un problème évident de succession et d'héritage. Deuxièmement, lorsque l'utilisateur choisit lui-même les mots de sa phrase mnémonique, l'entropie de la phrase est dramatiquement faible. Les êtres humains sont incapables de produire du hasard véritable : même une phrase qui semble aléatoire suit des schémas prédictibles (mots courants, citations, paroles de chansons...). Les attaquants exploitent ces biais avec des dictionnaires et des techniques de brute force, et de nombreux brainwallets ont été vidés quelques secondes après leur création.

Le seul scénario où un brainwallet peut se justifier est celui d'une fuite urgente (passage de frontière, checkpoint...), où aucun support physique ne peut être transporté sans risque. Mais même dans ce cas, des alternatives moins risquées existent : dissimulation physique de la phrase, stéganographie, chiffrement...

Termes associés : **ENTROPIE** | **PHRASE DE RÉCUPÉRATION**

BRANCH-AND-BOUND

PORTEFEUILLE

Méthode utilisée pour la sélection de pièces dans le portefeuille de Bitcoin Core depuis la version 0.17 et inventée par Murch. Le BnB est une recherche pour trouver un ensemble d'UTXO qui correspond au montant exact des sorties à satisfaire dans une transaction, afin de minimiser le change et les frais associés. Son but est de réduire un critère de gaspillage qui prend en compte à la fois le coût immédiat et les coûts futurs prévus pour le change. Cette méthode est plus précise en termes de frais par rapport aux heuristiques antérieures comme le *Knapsack Solver*. Le *Branch-and-Bound* est inspiré de la méthode de résolution de problème de même nom, inventée en 1960 par Ailsa Land et Alison Harcourt.

Cette méthode est également parfois nommée « Algorithme de Murch » en référence à son inventeur.

Termes associés : **KNAPSACK SOLVER** | **UTXO**

BRANCHE - BITCOIN

PROTOCOLE | EN : BRANCH - BITCOIN

Dans le cadre de Bitcoin, une branche est une suite de blocs valides dans laquelle chaque nouveau bloc est lié au précédent. La blockchain représente la branche qui contient le plus de travail accumulé, mais il peut exister des branches concurrentes résultant d'embranchements. Chaque branche partage au moins un bloc commun avec la branche principale, à savoir le bloc de Genèse. La partie des blocs propres à la branche est parfois appelée le « segment ». Lorsque deux branches concurrentes sont présentes, les nœuds honnêtes se synchronisent sur celle qui dispose du plus de travail accumulé. Une branche est parfois qualifiée de « forte » si elle a plus de travail accumulé qu'une autre, et de « faible » dans le cas contraire.

Termes associés : **BLOC** | **FORK**

BRANCHE - GIT

INFORMATIQUE | EN : BRANCH - GIT

Dans le cadre de Git, représente une séparation du flux de travail principal, permettant le développement en parallèle, sans affecter la branche principale (généralement nommée `master` ou `main`). Les branches permettent les modifications, les tests et les expérimentations dans un environnement isolé, avant leur éventuelle intégration dans le projet principal via un *merge* (fusion).

Termes associés : **GIT** | **MERGE**

BRC-20

COUCHE SUPÉRIEURE

Standard permettant l'interaction avec des jetons non natifs sur Bitcoin, basé sur les inscriptions du protocole Ordinals. Créé par Domo en mars 2023, ce standard expérimental a suscité un fort engouement spéculatif, entraînant une hausse historique des frais de transaction en avril et mai 2023. Les jetons BRC-20 utilisent Bitcoin pour stocker et horodater des fonctions JSON gérant ces actifs. Les fonctions sont `deploy` (création de jetons), `mint` (émission de jetons) et `transfer` (transfert de jetons). Pour exécuter ce protocole, il faut que des personnes maintiennent des serveurs qui recensent l'intégralité des fonctions. Le standard BRC-20 est alors une utilisation très peu optimisée de Bitcoin par rapport à d'autres protocoles de création de jetons tels que RGB.

Termes associés : **ORDINALS THEORY** | **RGB**

BREEZ

LIGHTNING NETWORK

Portefeuille mobile Lightning Network non-custodial qui permet d'envoyer et de recevoir des paiements en bitcoins de manière quasi instantanée. Breez intègre un nœud Lightning directement sur l'appareil de l'utilisateur, lui offrant le contrôle total de ses fonds sans dépendre d'un tiers de confiance.

L'application se distingue par son interface simplifiée qui masque la complexité technique du Lightning Network, notamment la gestion des canaux de paiement et de la liquidité. Breez intègre également des fonctionnalités de point de vente, permettant aux petits commerçants d'accepter des paiements Lightning directement depuis leur téléphone. L'entreprise propose aussi le Breez SDK, un kit de développement open-source qui permet à d'autres applications d'intégrer des paiements Lightning en s'appuyant sur l'implémentation Greenlight de Core Lightning. Breez prend en charge les protocoles LNURL et Lightning Address pour simplifier la réception de paiements.

Termes associés : **LIGHTNING NETWORK** | **PHOENIX**

BROLLUPS

COUCHE SUPÉRIEURE

Concept de rollup natif sur Bitcoin proposé par Burak en juin 2024. Contrairement aux rollups optimistes ou à base de preuves à divulgation nulle de connaissance, les Brollups forment une catégorie à part. Ils s'appuient sur le concept de VTXO (*Virtual UTXO*), issu du protocole Ark, et les utilisent comme mécanisme d'ancrage (*peg*). Le principe consiste à échanger des VTXOs contre des données d'appel (*calldata*), accompagnées de nouveaux VTXOs, via des coinswaps. Les Brollups sont opérés par un opérateur ou un quorum d'opérateurs qui fournissent la liquidité au protocole et font avancer l'état du rollup en enchaînant des transactions Bitcoin à intervalles réguliers. La blockchain Bitcoin sert de couche de disponibilité des données, tandis que les transactions sont exécutées sur une machine virtuelle dédiée appelée *Bitcoin Virtual Machine*. L'ancrage est exécutable on-chain et peut être racheté unilatéralement à tout moment. Les Brollups ne nécessitent aucune modification du protocole Bitcoin

et ne sont associés à aucun jeton.

Termes associés : ARK | ROLLUP

BSMS

PORTEFEUILLE

Sigle de « *Bitcoin Secure Multisig Setup* ». BSMS est un protocole standardisé par le BIP-0129 qui permet de créer un portefeuille multisignatures de manière sécurisée entre plusieurs signataires (*hardware wallets*, logiciels), en garantissant que tous s'accordent sur la même configuration avant de l'utiliser.

Le protocole fonctionne en deux tours d'échanges entre un Coordinateur et les Signataires. Au premier tour, le Coordinateur définit les paramètres du multisig (seuil M-de-N, type de script) et distribue éventuellement un jeton de chiffrement (`TOKEN`) à chaque signataire. Chaque signataire fournit alors sa clé publique étendue accompagnée de son chemin de dérivation et d'une signature prouvant qu'il contrôle la clé privée correspondante. Au second tour, le Coordinateur rassemble toutes les clés, génère un descripteur (*output descriptor*) du portefeuille et le redistribue à chaque signataire. Chaque signataire vérifie que sa propre clé est bien incluse et que la première adresse du portefeuille est correcte, ce qui sert de somme de contrôle sur l'ensemble de la configuration.

Trois modes de chiffrement sont possibles pour les échanges : aucun chiffrement, un chiffrement standard avec un jeton partagé de 64 bits, ou un chiffrement étendu avec un jeton unique de 128 bits par signataire. Le chiffrement utilise AES-256-CTR avec une clé dérivée via PBKDF2. Le protocole BSMS est notamment implémenté dans Nunchuk, Sparrow et ColdCard.

Termes associés : BIP-0129 | MULTISIG

BSV - BITCOIN SATOSHI VISION

HISTOIRE

Système de cryptomonnaie issu d'un hard fork de Bitcoin Cash (BCH), lui-même un fork de Bitcoin (BTC). Le fork de Bitcoin SV s'est produit le 15 novembre 2018 au bloc 556 766 en raison de désaccords au sein de la communauté Bitcoin Cash, notamment concernant la taille des blocs et la supposée vision de Satoshi Nakamoto, le créateur de Bitcoin. Deux camps se sont affrontés :

- Les partisans du fork « Bitcoin Cash ABC », qui est devenu Bitcoin Cash (BCH). Ce groupe était notamment soutenu par le célèbre entrepreneur Roger Ver ;
- Les promoteurs du fork « Bitcoin Cash Satoshi Vision », qui a abouti à la création de Bitcoin Satoshi Vision (BSV). Ce camp était entre autres soutenu par Craig Wright.

Bitcoin SV se distingue de Bitcoin Cash, et encore plus de Bitcoin, par sa limite de taille de bloc considérablement élevée. Cette spécificité vise à s'aligner sur ce que ses partisans considèrent être la vision initiale de Satoshi Nakamoto pour Bitcoin.

Bitcoin SV

Termes associés : BCH - BITCOIN CASH | BLOCKSIZE WAR

BTC

PROTOCOLE

Symbole boursier ou monétaire (*ticker*) utilisé pour représenter une unité de bitcoin sur les plateformes d'échange. Il sert à identifier rapidement le bitcoin parmi d'autres actifs et monnaies. Une unité de bitcoin (1 BTC) est égale à 100 000 000 de satoshis (ou « sats »).

« BTC » est le symbole historique du bitcoin, mais il ne respecte pas la norme ISO 4217, qui régit les codes monétaires. Selon cette norme, « BT » est assigné au Bhoutan et les monnaies non adossées à un État, telles que le bitcoin, devraient débuter par un « X ». C'est pourquoi certaines organisations utilisent le symbole « XBT » pour le bitcoin. Bien que « XBT » ne soit pas encore officiellement reconnu par la norme ISO 4217, il en respecte néanmoins les critères.

Terme associé : XBT

BTCD - BTC SUITE

PROTOCOLE

Implémentation minoritaire du protocole Bitcoin. BTCD est une alternative au logiciel majoritaire Bitcoin Core (notamment Bitcoind). Contrairement à Bitcoin Core, qui est développé en C++, BTCD est codé en Go (golang). Son objectif est de respecter les règles de consensus établies par Bitcoin Core pour assurer une compatibilité avec les autres nœuds du réseau. BTCD est développé dans le cadre du projet BTC Suite.

Termes associés : BITCOIN CORE | BITCOIND

BTCMAP

OUTIL

Carte interactive open source qui recense les commerces et services acceptant le bitcoin à travers le monde. BTCMap s'appuie sur les données d'OpenStreetMap et permet à quiconque de signaler ou de mettre à jour les points de vente qui acceptent les paiements en bitcoins, que ce soit on-chain ou via le Lightning Network.

L'outil est accessible via un site web et permet de rechercher des commerces par localisation géographique ou par catégorie (restaurants, hôtels, boutiques). BTCMap contribue à rendre visible l'écosystème des commerçants qui acceptent le bitcoin. Chaque point de vente peut être vérifié par la communauté, et les informations sont horodatées pour indiquer la date de la dernière vérification.

Termes associés : BTCPAY SERVER | LIGHTNING NETWORK

BTCPAY SERVER

OUTIL

Processeur de paiement open-source qui permet aux commerçants et aux utilisateurs d'accepter des paiements en bitcoins sans dépendre d'un tiers pour le traitement des transactions. Lancé en 2017, BTCPay Server offre une solution d'intégration de paiements en bitcoins pour les sites e-commerce, avec des fonctionnalités avancées comme le support de *hardware wallets*, des outils de facturation et de comptabilité, ainsi que la compatibilité avec le Lightning Network. Son développement a été initié par Nicolas Dorier, en réaction aux actions de BitPay qui, selon lui, avaient induit en erreur ses utilisateurs en les poussant vers l'adoption de SegWit2x, que la société considérait à tort comme le « vrai » Bitcoin. Cette opposition s'est cristallisée dans un tweet désormais célèbre de Nicolas Dorier en août 2017 :

« *This is lies, my trust in you is broken, I will make you obsolete* ».

Termes associés : LIGHTNING NETWORK | SEGWIT2X

BTG - BITCOIN GOLD

HISTOIRE

Système de cryptomonnaie créé à partir d'un *hard fork* de Bitcoin (BTC), qui a eu lieu le 24 octobre 2017 au bloc 491 407. Le fork a été initié par un groupe de développeurs et d'investisseurs mécontents de la direction prise par Bitcoin, en particulier en ce qui concerne la concentration croissante de la puissance de minage entre les mains de quelques grandes fermes utilisant des ASIC. Bitcoin Gold voulait démocratiser le processus de minage en utilisant un nouvel algorithme de preuve de travail, nommé *Equihash*, qui est résistant aux ASIC et favorise donc le minage par des cartes graphiques (GPU). Le but était de rendre la participation au minage accessible à un plus grand nombre de personnes afin de le décentraliser et de réduire les risques liés à la centralisation. BTG utilisait initialement le même algorithme de minage que Zcash (ZEC), à savoir Equihash avec les paramètres (200,9). En juin 2018, ils l'ont modifié pour adopter Equihash-BTG, une variante avec les paramètres (144,5), afin de renforcer la résistance aux ASIC.

Attention, l'altcoin Bitcoin Gold (BTG) ne doit pas être confondu avec bit gold, le concept de Nick Szabo.

Termes associés : ASIC | BIT GOLD

BTKN

COUCHE SUPÉRIEURE

Standard de jetons (*tokens*) natif du protocole Spark, permettant l'émission et la gestion d'actifs numériques directement sur une couche supérieure de Bitcoin. Les BTKN peuvent représenter des stablecoins, des jetons de fidélité, des points de récompense ou tout autre actif tokenisé. Contrairement aux protocoles de jetons sur d'autres blockchains, les BTKN ne nécessitent pas de smart contracts ni de déploiement complexe : l'émetteur utilise le SDK de Spark pour

créer un jeton, en émettre l'offre et effectuer des transferts.

Termes associés : **SPARK** | **STABLECOIN**

BTRUST

ORGANISATION

Organisation à but non lucratif fondée en février 2021 par Jack Dorsey et Jay-Z avec une dotation initiale de 500 bitcoins. Btrust a pour mission de décentraliser le développement des logiciels Bitcoin en encourageant et en finançant des développeurs dans les pays du Sud global, notamment en Afrique et en Inde. L'organisation propose des programmes de formation, des bourses de développement et des subventions pour renforcer la contribution de ces régions au code open source de Bitcoin et à son écosystème technique.

Terme associé : **BITCOIN CORE**

BUCKETING

RÉSEAU | EN : BUCKETING

Méthode de répartition des adresses de pairs dans les tables du module addrman de Bitcoin Core. Chaque adresse est assignée à un seau (« bucket ») par un calcul de hachage qui combine l'adresse elle-même, le groupe réseau dont elle provient et une clé secrète de 256 bits (`nKey`) propre à chaque nœud. Ce placement déterministe mais imprévisible de l'extérieur vise à empêcher un attaquant de prédire ou de contrôler quelles positions il occupe dans les tables.

Les tables « new » et « tried » utilisent des schémas de bucketing distincts. Dans la table « new », le seau dépend du groupe réseau source et du groupe réseau de l'adresse. Dans la table « tried », il dépend uniquement du groupe réseau de l'adresse elle-même. Cette séparation des critères limite la capacité d'un attaquant à monopoliser les seaux depuis une seule source, ce qui constitue une défense contre les attaques de type éclipse.

Termes associés : **ADDRMAN** | **ECLIPSE** | **PEERS.DAT**

BULL BITCOIN

ORGANISATION

Plateforme d'échange de bitcoins canadienne fondée par Francis Pouliot. Bull Bitcoin se distingue par son modèle *non-custodial* : contrairement aux plateformes d'échange traditionnelles, elle n'a pas la garde des bitcoins de ses clients. Les achats sont directement envoyés dans le portefeuille personnel de l'utilisateur, et les ventes sont effectuées depuis ce même portefeuille.

L'entreprise adopte une philosophie strictement « Bitcoin only » et ne propose aucune autre cryptomonnaie. Bull Bitcoin prend en charge les paiements via le Lightning Network pour les transactions de petits montants. La plateforme est particulièrement populaire au Canada et s'est étendue à d'autres marchés. Son modèle *non-custodial* réduit le risque de contrepartie pour les utilisateurs, car la plateforme ne détient jamais les fonds, ce qui la rapproche davantage des principes de souveraineté financière promus par Bitcoin.

Terme associé : **LIGHTNING NETWORK**

BULL MARKET

ÉCONOMIE ET RÉGULATION | FR : MARCHÉ HAUSSIER

Période prolongée durant laquelle le prix d'un actif, tel que le bitcoin, monte par rapport à une monnaie fiat, typiquement le dollar américain. L'expression « Bull Run », d'un registre plus familier, est souvent utilisée pour désigner ce même phénomène.

Termes associés : **BEAR MARKET** | **BULL RUN**

BULL RUN

ÉCONOMIE ET RÉGULATION | FR : MARCHÉ HAUSSIER

Période prolongée durant laquelle le prix d'un actif, tel que le bitcoin, augmente par rapport à une monnaie fiat, le plus souvent le dollar américain. L'expression « Bull Run » désigne le même phénomène que « Bull Market », mais dans un registre plus familier.

Termes associés : **BEAR MARKET** | **BULL MARKET**

BULLETPROOFS

CRYPTOGRAPHIE

Système de preuves à divulgation nulle de connaissance (*zero-knowledge proofs*) utilisé notamment dans les *Confidential Transactions* pour vérifier qu'un montant engagé dans une transaction se situe dans un intervalle valide, sans révéler sa valeur. Les Bulletproofs ont été introduits en 2017 par Bünz, Bootle, Boneh, Poelstra, Wuille et Maxwell. Ils remplacent les preuves de portée antérieures (*Borromean ring signatures*) en produisant des preuves significativement plus compactes, dont la taille croît de manière logarithmique ($O(\log n)$) par rapport au nombre de bits représentant la valeur.

Une évolution nommée Bulletproofs++, proposée par Liam Eagen, Sanket Kanjalkar, Tim Ruffing et Jonas Nick, améliore encore l'efficacité en termes de génération et de vérification. Sur Liquid, l'adoption des Bulletproofs a permis de réduire la taille des transactions confidentielles, diminuant ainsi les frais et ralentissant la croissance de la chaîne.

Termes associés : **CONFIDENTIAL TRANSACTIONS** | **ZKP - ZERO-KNOWLEDGE PROOF**

BURN ADDRESS

PROTOCOLE | FR : ADRESSE DE BRÛLAGE

Adresse Bitcoin vers laquelle des fonds sont envoyés de manière irréversible, sans possibilité de les dépenser par la suite. Personne ne possède la clé privée correspondante : envoyer des bitcoins vers une telle adresse revient à les détruire définitivement, à moins qu'il soit un jour possible de casser la cryptographie sous-jacente.

Ces adresses sont généralement construites de manière arbitraire, sans être dérivées d'une clé privée, afin de pouvoir prouver que l'on ne la connaît pas. La probabilité qu'une clé privée valide corresponde à une adresse choisie aléatoirement est astronomiquement faible, ce qui rend les fonds effectivement irrécupérables. Par exemple, il y a l'adresse :

`1BitcoinEaterAddressDontSendf59kuE`.

Le brûlage de bitcoins peut servir à différentes fins. Le protocole Counterparty l'a utilisé en 2014 lors de sa phase de *proof of burn* pour distribuer ses propres tokens en échange de bitcoins détruits. Cela peut également être une preuve d'engagement irréversible dans certains mécanismes cryptographiques, ou bien simplement une action ludique.

Terme associé : **ADRESSE DE RÉCEPTION**

BUSINESS LOGIC

RGB | FR : LOGIQUE MÉTIER

Dans le cadre du protocole RGB, la *Business Logic* regroupe l'ensemble des règles et des opérations internes d'un contrat, décrites par son schéma (c'est-à-dire la structure même du contrat). Elle dicte la manière dont l'état du contrat peut évoluer et sous quelles conditions.

Termes associés : **SCHEMA** **STATE TRANSITION**

C

C++

</> INFORMATIQUE

Langage de programmation polyvalent, évoluant du C, connu pour sa puissance et sa flexibilité. Utilisé pour le développement logiciel complexe, il prend en charge la programmation orientée objet et offre de riches fonctionnalités pour la gestion de la mémoire et des ressources système.

Bitcoin Core, l'implémentation majoritaire du protocole Bitcoin, est écrite en C++. C'est un héritage du tout premier client de Satoshi qui était aussi dans ce langage de programmation. Pour garantir une cohérence stricte du comportement des nœuds sur le réseau, notamment pour éviter les *forks*, il a été choisi de ne pas changer de langage depuis 2009.

Termes associés : **BITCOIN CORE** **FORK**

CAHOOTS

PORTEFEUILLE

Dans le cadre du portefeuille Samourai Wallet et d'autres logiciels de portefeuilles qui l'implémentent, un Cahoot désigne tous les types de transactions réalisées en collaboration entre plusieurs utilisateurs. Procéder à un Cahoot signifie donc participer conjointement à une transaction. Cette collaboration s'articule autour de l'échange de transactions partiellement signées. Ces échanges peuvent se faire soit manuellement, via des codes QR, soit de manière automatisée, via le réseau de communication *Soroban*. Parmi les cahoots, on retrouve : *Les transactions* Stowaway *(*Payjoin*)* ; *Les transactions* Stonewall x2* ; *Les transactions* Joinbot*.

Termes associés : **PAYJOIN** **STONEWALL X2**

CAKE WALLET

PORTEFEUILLE

Portefeuille open source disponible sur iOS, Android, macOS, Linux et Windows, initialement conçu pour Monero puis étendu à Bitcoin et d'autres cryptomonnaies. CakeWallet se distingue dans l'écosystème Bitcoin par son intégration de fonctionnalités de confidentialité, notamment le support des *Silent Payments* et de *Payjoin v2*. Le portefeuille est non-custodial et permet à l'utilisateur de conserver le contrôle de ses clés privées. Il propose un service d'échange intégré pour convertir entre différentes cryptomonnaies, ainsi qu'une intégration native de Tor pour renforcer la confidentialité au niveau du réseau.

Termes associés : **COINJOIN** **PORTEFEUILLE**

CANAAN

↗ MINAGE

Entreprise fondée en 2013 par Nangeng Zhang, spécialisée dans la fabrication de machines de minage de bitcoins. Canaan est à l'origine de l'AvalonMiner, le tout premier ASIC conçu spécifiquement pour le minage de Bitcoin. Basée à Singapour, Canaan est devenue en 2019 la première entreprise du secteur des ASIC de minage à être cotée au NASDAQ.

Termes associés : **ASIC** **MINAGE**

CANAL DE PAIMENT

↯ LIGHTNING NETWORK | EN : PAYMENT CHANNEL

Dans le cadre du Lightning Network, un canal de paiement est une connexion bidirectionnelle entre deux nœuds Lightning qui permet de faire des échanges de bitcoins off-chain. On-chain, un canal de paiement est représenté par une adresse multisignatures 2/2 détenue par les deux participants. Le canal de paiement nécessite une transaction on-chain pour son ouverture et une autre pour sa fermeture. Entre ces deux évènements, les utilisateurs du canal peuvent réaliser un très grand nombre d'échanges de bitcoins off-chain, sur le Lightning Network, sans nécessiter une activité on-chain. Sur Lightning, il est possible de router un paiement à travers plusieurs canaux et plusieurs nœuds, afin d'envoyer des bitcoins sans forcément ouvrir un canal direct avec le receveur.

Termes associés : **MULTISIG** **OFF-CHAIN**

CANDIDAT - BLOC

↗ MINAGE | EN : CANDIDATE BLOCK

Un bloc candidat est un bloc en cours de création par un mineur participant au processus de minage du système Bitcoin. Le bloc candidat est une structure de données temporaire qui contient des transactions en attente d'être confirmées, mais ne dispose pas encore d'une preuve de travail valide (*Proof-of-Work*) pour être ajouté à la blockchain. Le mineur sélectionne les transactions à inclure dans le bloc candidat en fonction de divers facteurs, tels que les frais de transaction associés et les contraintes de taille de bloc. Une fois les transactions sélectionnées, le mineur génère l'entête du bloc, qui comprend la version, un condensat des transactions (racine de Merkle), un horodatage, le hash du bloc précédent, la cible de difficulté et un nonce. Le mineur tente ensuite de trouver un hash de son entête satisfaisant la difficulté cible du moment. Pour ce faire, il modifie le nonce présent dans l'entête. Il peut également modifier d'autres informations présentes dans son bloc candidat. C'est le mécanisme de la preuve de travail. Si le mineur réussit à trouver un hash valide, le bloc candidat devient un bloc valide et est diffusé au réseau pour être ajouté à la blockchain.

Termes associés : **NONCE** **PREUVE DE TRAVAIL**

CAPACITÉ DE CANAL LIGHTNING

LIGHTNING NETWORK | EN : LIGHTNING CHANNEL CAPACITY

Quantité de bitcoins bloqués sur une adresse multisignatures qui représente un canal de paiement sur le Lightning Network. La capacité d'un canal est donc la quantité maximale de sats qui peut être transmise via ce canal spécifique. Elle est définie au moment de la création du canal par la somme des fonds que les parties engagent dans le canal. L'« *inbound capacity* », ou « capacité entrante », désigne la quantité maximale de bitcoins qu'un nœud peut recevoir via un canal. L'« *outbound capacity* », ou « capacité sortante » représente la quantité maximale de bitcoins qu'un nœud peut envoyer à travers un canal spécifique.

Termes associés : **CANAL DE PAIMENT** | **MULTISIG**

CASA

ORGANISATION

Entreprise américaine proposant des solutions de sécurisation de bitcoins en multisig pour les particuliers. Casa offre un service où les clés privées sont réparties entre l'utilisateur (sur son *hardware wallet* et son téléphone) et Casa (qui détient une clé de récupération d'urgence). Ce modèle en 2-de-3 ou 3-de-5 permet de sécuriser ses fonds sans point unique de défaillance, tout en disposant d'une option de récupération en cas de perte d'un appareil.

Casa propose également un protocole d'héritage intégré. L'utilisation se fait via une application mobile et le service fonctionne sur abonnement avec différents niveaux de sécurité.

Termes associés : **HARDWARE WALLET** | **MULTISIG**

CASASCIUS

HISTOIRE

Pièces physiques qui contiennent des bitcoins, créées à partir de 2011 par Mike Caldwell. Chaque pièce Casascius intègre une clé privée dissimulée sous un hologramme inviolable, qui permet à son détenteur d'accéder aux bitcoins associés.

Les pièces étaient disponibles en plusieurs dénominations (1 BTC, 10 BTC, 25 BTC, 100 BTC...) et arborent un design avec un « B » doré. Elles permettaient de transférer physiquement des bitcoins sans passer par le réseau, en remettant simplement la pièce au destinataire.

En novembre 2013, le Financial Crimes Enforcement Network (FinCEN) a considéré cette activité comme une forme de transmission monétaire qui nécessite un enregistrement. Mike Caldwell a alors cessé de vendre des pièces préchargées, tout en continuant à produire des pièces vierges. Les pièces Casascius sont devenues des objets de collection recherchés.

Termes associés : **BITCOIN CORE** | **E-GOLD**

CASHU

COUCHE SUPÉRIEURE

Protocole open-source de monnaie électronique chaumienne, similaire au système eCash de David Chaum, mais qui fonctionne sur Bitcoin et le Lightning Network. Plus précisément, Cashu est inspiré d'une variante d'eCash proposée en 1996 par David Wagner nommée « *Chaumian ecash without RSA* ». Cashu peut être utilisé sur des portefeuilles custodiaux afin que le serveur ne puisse identifier ni les propriétaires des fonds, ni les détails des transactions, offrant ainsi une amélioration de la confidentialité. Les utilisateurs peuvent générer des jetons Cashu en échange de bitcoins, qui sont signés par le serveur sans connaître l'utilisateur. Les jetons peuvent ensuite être transférés entre utilisateurs de manière instantanée, privée et sans frais.

Terme associé : **ECASH - DAVID CHAUM**

CBDC

ÉCONOMIE ET RÉGULATION | FR : MONNAIE NUMÉRIQUE DE BANQUE CENTRALE

Sigle de « *Central Bank Digital Currency* » (monnaie numérique de banque centrale). Il s'agit d'une forme numérique de monnaie fiat émise et contrôlée directement par une banque centrale, contrairement aux dépôts bancaires classiques qui sont créés par les banques commerciales.

Les projets de CBDC se développent dans de nombreux pays : l'euro numérique en Europe, le yuan numérique (e-CNY) en Chine, ou le *Sand Dollar* aux Bahamas. Elles peuvent fonctionner sur des registres centralisés ou utiliser des technologies inspirées des blockchains, mais restent sous le contrôle total de l'autorité émettrice.

Les partisans y voient un outil de modernisation des paiements et d'inclusion financière. Les critiques, notamment dans la communauté Bitcoin, soulèvent des risques majeurs pour la vie privée : une CBDC permettrait potentiellement à l'État de surveiller, bloquer ou conditionner chaque transaction des citoyens, constituant un outil de contrôle financier sans précédent. Les CBDC sont fondamentalement différentes de Bitcoin, qui est décentralisé et résistant à la censure.

Terme associé : **FIAT**

CBTC - CITREA BITCOIN

COUCHE SUPÉRIEURE

Représentation native du bitcoin sur le rollup Citrea. Le cBTC (*Citrea Bitcoin*) est la devise native utilisée pour payer les frais de *gas* et effectuer des transactions sur le réseau Citrea. Il est obtenu en déposant des BTC sur la chaîne principale Bitcoin via le pont Clementine, qui verrouille les fonds et émet un montant équivalent en cBTC sur le rollup. Il peut être reconverti en BTC à tout moment via le mécanisme de *peg-out* de Clementine.

Termes associés : **CITREA** | **CLEMENTINE**

CENSORSHIP RESISTANCE

PROTOCOLE | FR : RÉSISTANCE À LA CENSURE

Propriété d'un système qui le rend capable de fonctionner sans qu'aucune entité ne puisse empêcher des participants légitimes de l'utiliser. Dans le contexte de Bitcoin, la résistance à la censure signifie qu'aucun gouvernement, entreprise ou individu ne peut bloquer une transaction valide ou interdire à un utilisateur d'accéder au réseau.

Cette propriété repose sur plusieurs caractéristiques fondamentales de Bitcoin : la décentralisation du réseau (des milliers de nœuds répartis dans le monde), l'absence d'autorité centrale pouvant filtrer les transactions, et le fonctionnement par preuve de travail qui rend l'inclusion des transactions dépendante uniquement des frais offerts par l'émetteur.

La résistance à la censure est considérée comme l'une des propositions de valeur les plus importantes de Bitcoin, particulièrement dans les régimes autoritaires où les citoyens peuvent voir leurs comptes bancaires gelés ou leurs transactions bloquées. Elle distingue fondamentalement Bitcoin des systèmes de paiement centralisés et des projets de CBDC, où une autorité peut à tout moment décider d'exclure des utilisateurs.

Termes associés : **DÉCENTRALISATION** | **MINAGE**

CET

COUCHE SUPÉRIEURE

Sigle de « *Contract Execution Transaction* ». C'est une transaction spécifique au sein d'un DLC qui permet le règlement final entre les parties en fonction de l'issue d'un événement futur. Lorsque l'oracle publie une signature correspondant au résultat de l'événement, les parties utilisent cette signature pour compléter et déverrouiller la CET qui envoie les fonds à la partie gagnante. La CET signée est ensuite diffusée sur le réseau, et le gagnant reçoit les bitcoins qui lui sont dus selon les conditions du contrat intelligent. Toutes les autres CET potentielles, qui auraient été exécutées en cas de résultats différents, deviennent obsolètes et sont abandonnées.

Terme associé : **DLC - DISCREET LOG CONTRACT**

CGMINER

MINAGE

Logiciel de minage *open-source* initialement développé pour le CPU et le GPU. CGminer permet aux utilisateurs de gérer et d'optimiser leurs opérations de minage, avec des fonctionnalités comme le *multi-threading*, la gestion de pools multiples et la surveillance en temps réel des performances. Le logiciel n'est plus vraiment maintenu de nos jours, mais il a été remplacé par le projet *open-source* BOSminer de Braiins, qui a été réécrit de zéro en Rust.

Terme associé : **BOSMINER**

CHACHA20-POLY1305

CRYPTOGRAPHIE

Algorithme de chiffrement authentifié avec données associées qui combine le chiffre de flux ChaCha20 et le code d'authentification de message Poly1305. Il prend en entrée une clé de 256 bits et un nonce de 96 bits, et produit un texte chiffré accompagné d'une étiquette d'authentification de 128 bits qui garantit à la fois l'intégrité et l'authenticité des données.

Dans l'écosystème Bitcoin, ChaCha20-Poly1305 est utilisé dans le protocole de transport P2P v2 (BIP-0324) pour chiffrer les communications entre nœuds du réseau. Il est également utilisé dans la couche de transport du Lightning Network (BOLT-08), où le framework Noise l'utilise pour chiffrer et authentifier les messages échangés entre pairs après le *handshake* initial.

Termes associés : **NOISE** **P2P TRANSPORT V2**

CHAINALYSIS

ORGANISATION

Entreprise américaine fondée en 2014, spécialisée dans l'analyse de la blockchain et le traçage des transactions de cryptomonnaies. Chainalysis fournit des logiciels et des services aux gouvernements, aux forces de l'ordre, aux institutions financières et aux plateformes d'échange pour détecter et suivre les flux de fonds sur les blockchains publiques, notamment celle de Bitcoin. Ses outils permettent de suivre des fonds et d'identifier des liens entre des transactions et des entités réelles. Pour ce faire, Chainalysis exploite des heuristiques d'analyse de chaîne.

Terme associé : **COINJOIN**

CHAINCODE LABS

ORGANISATION

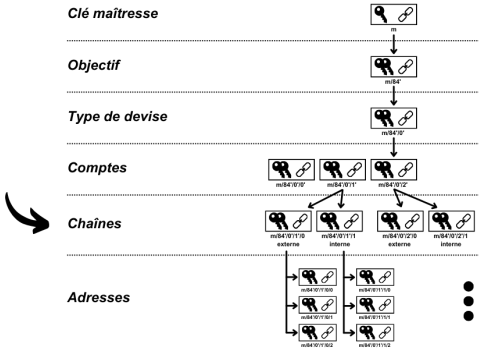
Centre de recherche et développement sur Bitcoin fondé en 2014 par Alex Morcos et Suhas Daftuar à New York. Chaincode Labs finance et emploie des développeurs de Bitcoin Core, et organise des programmes de formation pour les contributeurs open source au protocole Bitcoin.

Terme associé : **BITCOIN CORE**

CHAINE EXTERNE

PORTEFEUILLE	EN : EXTERNAL KEYCHAIN

Dans la dérivation des portefeuilles déterministes et hiérarchiques, la chaîne externe est une branche de dérivation utilisée pour générer des adresses de réception destinées à recevoir des paiements venus de l'extérieur, c'est-à-dire d'un autre portefeuille. Chaque compte tel que défini en profondeur 3 dispose de deux chaînes en profondeur 4 : une chaîne externe et une chaîne interne (également appelée « change »). La chaîne externe est dérivée avec un index de `/0/`. La chaîne externe dérive des adresses destinées à être communiquées publiquement, c'est-à-dire les adresses que l'on nous propose lorsque l'on clique sur le bouton « recevoir » dans notre logiciel de portefeuille.

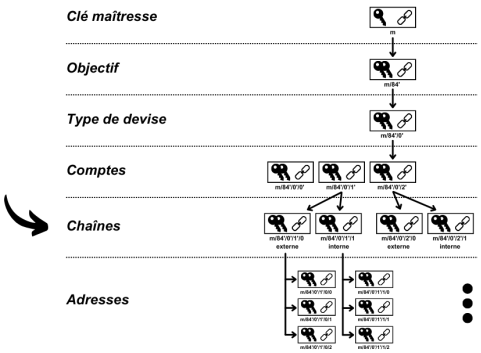


Terme associé : **CHEMIN DE DÉRIVATION**

CHAINE INTERNE

PORTEFEUILLE	EN : INTERNAL KEYCHAIN

Dans la dérivation des portefeuilles déterministes et hiérarchiques, la chaîne interne est une branche de dérivation utilisée pour générer des adresses de réception destinées à recevoir des paiements venus du même portefeuille, c'est-à-dire uniquement des adresses de change. Chaque compte tel que défini en profondeur 3 dispose de deux chaînes en profondeur 4 : une chaîne externe et une chaîne interne (également appelée « change »). La chaîne interne est dérivée avec un index de `/1/`.



Terme associé : **CHEMIN DE DÉRIVATION**

CHAINSPLIT

PROTOCOLE

Nom parfois donné à un embranchement naturel, c'est-à-dire une séparation temporaire de la blockchain résultant de la diffusion quasi simultanée de plusieurs blocs par différents mineurs à une même hauteur.

Termes associés : **EMBRANCHEMENT NATUREL** **FORK**

CHAINSTATE/

PROTOCOLE

Nom technique donné au dossier utilisé pour stocker l'UTXO set sur Bitcoin Core. C'est donc en réalité un synonyme d'« UTXO set ».

Terme associé : **UTXO SET**

CHAINWAY LABS

ORGANISATION

Entreprise à l'origine du développement de Citrea, le premier ZK rollup déployé sur Bitcoin. Fondée avec l'objectif de construire des solutions d'infrastructure pour Bitcoin, Chainway Labs est spécialisée dans les technologies de preuves à divulgation nulle de connaissance appliquées à l'écosystème Bitcoin. L'équipe développe notamment Citrea et son pont bidirectionnel Clementine, basé sur BitVM. L'ensemble du code produit par Chainway Labs est open source, publié sous le compte GitHub `chainwayxyz`.

Termes associés : **CITREA** **CLEMENTINE**

CHAINWORK

PROTOCOLE | FR : TRAVAIL CUMULÉ

Métrique qui quantifie le nombre total de hachages attendus pour produire une blockchain donnée. Le chainwork sert de critère pour déterminer la chaîne valide : les nœuds suivent toujours la chaîne qui dispose du plus grand chainwork, et non simplement celle qui contient le plus de blocs. Le calcul repose sur la cible de difficulté de chaque bloc. Pour un bloc donné, le nombre moyen de hachages nécessaires pour obtenir une empreinte inférieure ou égale à la cible est :

$$\frac{2^{256}}{cible\ +\ 1}$$

Le chainwork total d'une chaîne est la somme de ces valeurs pour chaque bloc, du bloc de genèse jusqu'au bloc considéré. Comme la difficulté varie au fil du temps, deux chaînes de longueurs identiques peuvent avoir un chainwork très différent : une chaîne minée sous une difficulté élevée aura un chainwork supérieur à une chaîne plus longue construite sous une difficulté artificiellement basse.

Dans la première version du logiciel, Satoshi Nakamoto utilisait le nombre de blocs comme critère de sélection de la chaîne valide. Cette approche était vulnérable à la manipulation, car un attaquant pouvait construire une chaîne plus longue en maintenant la difficulté à un niveau bas. Le chainwork a remplacé cette métrique pour garantir que la chaîne valide soit bien celle qui a nécessité le plus de travail réel. La valeur est accessible via la commande `bitcoin-cli getblock` dans le champ `chainwork`.

Termes associés : **DIFFICULTÉ** | **RESYNCHRONISATION**

CHAMPION

BIP | EN : CHAMPION - BIP

Dans le contexte du développement de Bitcoin Core (l'implémentation majoritaire de Bitcoin) le champion est la personne qui rédige et prend en charge la gestion d'un BIP (proposition d'amélioration de Bitcoin). Le champion est responsable de rédiger le BIP en suivant le processus formel, de le défendre auprès de la communauté, de répondre aux questions et de guider les discussions pour affiner la proposition. Il travaille également à obtenir le consensus nécessaire pour que le BIP soit accepté et implémenté. Il peut

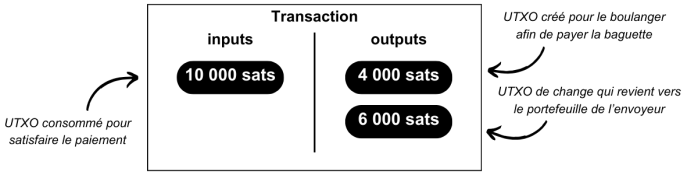
Le champion est également parfois nommé « auteur ». On retrouve d'ailleurs son nom noté derrière l'entête « Author » dans les BIPs. Il peut parfois y avoir plusieurs champions pour un même BIP.

Terme associé : **BIP**

CHANGE

PORTEFEUILLE

Dans le cadre des transactions Bitcoin, fait référence à l'UTXO créé avec les fonds restants après que le paiement effectif a été satisfait. Lorsque l'on utilise en entrées des UTXOs avec une quantité de bitcoins supérieure au montant nécessaire pour le paiement effectif et les frais de transaction, le surplus est un UTXO renvoyé à une adresse interne du portefeuille, appelée adresse de change. Le change représente cet UTXO. Par exemple, si vous souhaitez payer une baguette qui coute `4 000 sats` avec un UTXO de `10 000 sats`, vous allez créer dans votre transaction un change de `6 000 sats` (si l'on néglige les frais de transaction).



Même si c'est très peu utilisé, on pourrait également parler de « monnaie » (rendu de monnaie) pour évoquer le change.

Termes associés : **CHAINE INTERNE** | **UTXO**

CHANGE AVOIDANCE

CONFIDENTIALITÉ | FR : ÉVITEMENT DU CHANGE

Stratégie de gestion des UTXOs qui consiste à sélectionner les entrées d'une transaction de manière à éliminer ou minimiser le besoin de créer un output de change. L'objectif est de faire correspondre au plus près la somme des inputs avec le montant du paiement et les frais de transaction.

L'intérêt principal de cette approche est la confidentialité. Un output de change est souvent identifiable par les outils d'analyse de chaîne et constitue une information pour tracer des fonds dans le temps. En l'éliminant, on réduit les informations disponibles pour un observateur extérieur, on s'empêche de révéler des informations dans le futur via la CIOH, et on peut également induire en erreur l'analyse de chaîne.

Certains portefeuilles implémentent cette stratégie automatiquement dans leur algorithme de sélection de pièces.

Termes associés : **CHANGE** | **UTXO**

CHANNEL ACCEPTOR

LIGHTNING NETWORK

Mécanisme de LND permettant de définir une logique personnalisée pour accepter ou refuser automatiquement les demandes d'ouverture de canal entrantes. Lorsqu'un pair distant propose d'ouvrir un canal, la requête est transmise au *channel acceptor* qui répond par `TRUE` ou `FALSE`, accompagné éventuellement d'un message d'erreur à destination de l'initiateur.

Terme associé : **LND**

CHANNEL ANNOUNCEMENT

LIGHTNING NETWORK | FR : ANNONCE DE CANAL

Message diffusé sur le réseau Lightning pour signaler l'existence d'un nouveau canal de paiement aux autres nœuds. C'est ce message qui distingue un canal public d'un canal privé : seuls les canaux dont les partenaires choisissent de diffuser un *channel announcement* apparaissent dans le graphe du réseau et peuvent être utilisés par des tiers pour router des paiements.

Le message contient notamment les identifiants des deux nœuds partenaires, la référence à la transaction de financement on-chain, ainsi que les signatures des deux parties. À la réception de ce message, les autres nœuds du réseau vérifient que les partenaires possèdent bien les fonds dans la transaction de financement avant d'ajouter le canal à leur base de données locale. Le *channel announcement* fait partie du protocole de *gossip* du Lightning Network, qui assure la propagation des informations sur la topologie du réseau entre les pairs. Les canaux qui ne diffusent pas ce message sont dits « privés » (ou « non annoncés ») et restent invisibles dans le graphe.

Termes associés : **GOSSIP** | **PUBLIC CHANNEL**

CHANNEL BREACH

LIGHTNING NETWORK | FR : VIOLATION DE CANAL

Tentative de triche sur le Lightning Network, dans laquelle un pair publie une ancienne transaction d'engagement (*commitment transaction*) pour tenter de récupérer un solde qui ne lui revient plus. Lorsqu'un tel état révoqué est diffusé sur la blockchain, la contrepartie dispose d'un délai défini par le *timelock* pour détecter cette fraude et publier une transaction de justice qui lui permet de récupérer l'intégralité des fonds du canal.

La détection d'une violation de canal peut être assurée directement par le nœud s'il est en ligne, ou déléguée à une *watchtower* qui surveille la blockchain en permanence. Si la violation n'est pas contestée avant l'expiration du *timelock*, le tricheur conserve les fonds indûment récupérés.

Termes associés : **TRANSACTION D'ENGAGEMENT** | **WATCHTOWER**

CHANNEL.DB

LIGHTNING NETWORK

Base de données utilisée par LND pour stocker l'état complet de tous les canaux Lightning d'un nœud. Ce fichier est situé dans le répertoire `.lnd/data/graph/mainnet` et il contient la liste des canaux ouverts, les identifiants des pairs, les dernières transactions d'engagement pour chaque canal et les informations nécessaires pour punir un pair qui tenterait de diffuser un ancien état.

Le fichier `channel.db` évolue en permanence : chaque paiement, chaque routage, chaque ouverture ou fermeture de canal modifie son contenu. C'est pourquoi il ne constitue pas un support de sauvegarde fiable au sens classique. Restaurer une version obsolète de `channel.db` expose l'opérateur au mécanisme de pénalité du protocole Lightning : les pairs peuvent considérer qu'un ancien état est diffusé et récupérer l'intégralité des fonds du canal. La seule utilisation sûre de `channel.db` pour une restauration est la récupération directe depuis une machine qui vient de tomber en panne, ce qui garantit que l'état est bien le plus récent. Pour une sauvegarde fiable, LND propose plutôt le mécanisme de SCB (*Static Channel Backup*).

Termes associés : **SCB - STATIC CHANNEL BACKUP** | **TRANSACTION D'ENGAGEMENT**

CHANNEL FACTORIES

LIGHTNING NETWORK

Mécanisme avancé en cours de travail sur Lightning, permettant la création et la gestion de plusieurs canaux de paiement à partir d'un seul UTXO. Les channel factories utilisent des adresses multisig `n-of-n` pour qu'un groupe d'utilisateurs puisse détenir collectivement un seul UTXO. De là, ils peuvent ouvrir et fermer des canaux de paiement entre eux sans transactions supplémentaires on-chain, sauf lorsqu'ils souhaitent retirer leurs fonds de la factory. Cette méthode permettrait de réduire considérablement les coûts et l'espace occupé sur Bitcoin pour des transactions Lightning. En pratique, cela signifie que des opérations qui nécessiteraient normalement des transactions on-chain pour chaque ouverture ou fermeture de canal peuvent être effectuées hors chaîne, avec la sécurité garantie par la capacité de publier les transactions non publiées si nécessaire. Pour reprendre les mots de David A. Harding, les channel factories peuvent être décrites comme des canaux Lightning utilisés pour générer d'autres canaux Lightning.

Termes associés : **CANAL DE PAIMENT** | **MULTISIG**

CHANNEL JAMMING

ATTAQUE | FR : BROUILLAGE DE CANAL

Attaque par déni de service sur le Lightning Network qui consiste à bloquer la capacité de routage d'un ou plusieurs canaux de paiement pendant une durée prolongée, empêchant les nœuds honnêtes de router des paiements. L'attaquant s'envoie un paiement fictif à lui-même à travers une série de canaux intermédiaires, puis refuse de finaliser ou d'annuler le HTLC, ce qui immobilise les fonds des nœuds de routage sans frais pour l'attaquant.

L'attaque se décline en deux variantes : *Le* liquidity jamming* (décrit dès 2015 sous le nom de « loop attack ») exploite l'effet de levier du routage multi-sauts : avec x bitcoins, un attaquant peut verrouiller jusqu'à $20x$ bitcoins sur un chemin de 20 canaux. *Le* slot jamming* exploite la limite de 483 HTLC simultanés par canal : en envoyant 483 micro-paiements, un attaquant sature les emplacements disponibles avec un capital minimal.

Plusieurs pistes de mitigation sont explorées : frais payés à l'avance (*upfront fees*) par l'expéditeur, système de réputation locale entre pairs, certificats de preuve de fonds (*stake certificates*) et jetons de réputation. Le bLIP-0004 propose un mécanisme d'approbation des HTLC (*HTLC endorsement*) comme première mesure concrète. Aucune solution définitive n'a encore fait consensus.

Termes associés : **BLIP-0004** | **HTLC**

CHANNEL LEASE

LIGHTNING NETWORK | FR : LOCATION DE CANAL

Contrat d'ouverture de canal Lightning pour une durée déterminée, négocié via la plateforme Lightning Pool. Un acheteur de liquidité (*taker*) paie une prime à un vendeur (*maker*) pour que celui-ci ouvre un canal d'une capacité et d'une durée définies, exprimée en nombre de blocs. Le vendeur s'engage à maintenir le canal ouvert pendant toute la durée du bail.

Terme associé : **CHANNEL ACCEPTOR**

CHANNEL POINT

LIGHTNING NETWORK

Identifiant unique d'un canal Lightning sur la blockchain Bitcoin. Il se compose du TXID de la transaction de financement du canal suivi de l'index de la sortie correspondante, sous la forme `txid:output_index`. Par exemple : `a3f2…b7c:0` désigne la première sortie de la transaction de financement. Le *channel point* permet de retrouver la transaction on-chain qui a établi le canal et sert de référence dans les commandes d'administration du nœud, par exemple pour fermer un canal spécifique ou consulter ses détails.

Termes associés : **FORCE CLOSE** | **LND**

CHANNEL RESERVE

LIGHTNING NETWORK | FR : RÉSERVE DE CANAL

Montant minimal que chaque partie d'un canal Lightning doit conserver de son côté et ne peut pas dépenser. Cette réserve garantit que chaque participant a toujours quelque chose à perdre en cas de tentative de fraude, ce qui rend les transactions de pénalité économiquement dissuasives.

Lors de l'ouverture d'un canal, chaque nœud impose à son pair une *channel reserve*, généralement fixée à 1 % de la capacité du canal (avec un plancher de 546 satoshis, correspondant au *dust limit*). Par exemple, dans un canal de 1 000 000 sats, chaque partie doit maintenir au moins 10 000 sats de son côté. Ce seuil est défini dans le champ `channel_reserve_satoshis` lors de la négociation du canal (BOLT-02).

La *channel reserve* réduit légèrement la capacité effectivement utilisable d'un canal. Dans le protocole v1, elle peut être asymétrique, chaque côté pouvant imposer un montant différent. Dans le cas du *dual funding* (v2), la réserve est fixée à 1 % du solde total du canal pour les deux parties et n'est plus négociable.

Termes associés : **CANAL DE PAIMENT** | **TRANSACTION D'ENGAGEMENT**

CHANTOOLS

OUTIL

Outil en ligne de commande développé par Lightning Labs permettant de récupérer des fonds bloqués dans des canaux Lightning lorsque le nœud LND ne fonctionne plus correctement. Chantools intervient dans les situations d'urgence où les mécanismes de récupération standards (restauration par *Static Channel Backup*) échouent ou ne suffisent pas.

L'outil propose une trentaine de commandes couvrant différents scénarios de récupération : extraction d'un fichier `channel.backup` à partir d'une base de données corrompue (`chanbackup`), publication forcée du dernier état connu d'un canal (`forceclose`), balayage des fonds après expiration du timelock (`sweeptimelock`), récupération de fonds issus de fermetures forcées par le pair distant (`sweepremoteclosed`), ou encore récupération coopérative de canaux zombies (`zombierecovery`). Certaines commandes sont également compatibles avec Core Lightning. L'utilisation de chantools nécessite la seed du nœud, mais celle-ci ne quitte jamais la machine locale.

Termes associés : **FORCE CLOSE** | **LND**

CHARGE UTILE

INFORMATIQUE | EN : PAYLOAD

Dans le contexte général de l'informatique, une charge utile désigne les données essentielles transportées dans un paquet de données plus large. Par exemple, dans une adresse SegWit V0 sur Bitcoin, la charge utile correspond au hachage de la clé publique, sans les diverses métadonnées (le HRP, le séparateur, la version de SegWit et la somme de contrôle). Par exemple, sur l'adresse `bc1qc2eukw7reasfcmrafevp5dhv8635yuqays50gj`, nous avons :

- `bc` : la partie lisible par l'homme (HRP) ;

- `1` : le séparateur ;
- `q` : la version de SegWit. Ici, c'est la version 0 ;
- `c2eukw7reasfcmrafevp5dhv8635yuqa` : la charge utile, ici, le hachage de la clé publique ;
- `ys50gj` : la somme de contrôle.

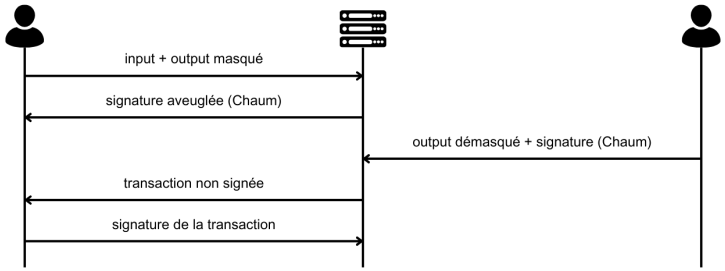
Termes associés : **CLÉ PUBLIQUE** | **SEGWIT V0**

CHAUMIAN COINJOIN

CONFIDENTIALITÉ | FR : COINJOIN CHAUMIEN

Protocole de coinjoin qui utilise les signatures aveugles de David Chaum et Tor pour les communications entre les participants et le serveur du coordinateur. L'objectif d'un coinjoin chaumien est de garantir aux participants que le coordinateur ne peut ni voler les bitcoins, ni faire de liens entre les inputs et les outputs.

Pour ce faire, les utilisateurs soumettent leur input et une adresse de réception cryptographiquement aveuglée au coordinateur. Cette adresse, une fois démasquée, sera destinée à recevoir les bitcoins en output de coinjoin. Le coordinateur signe ces tokens et les renvoie aux utilisateurs. Les utilisateurs se reconnectent ensuite de manière anonyme au serveur du coordinateur avec une nouvelle identité Tor et révèlent ensuite leurs adresses de sortie en clair pour la construction de la transaction. Le coordinateur peut vérifier que toutes ces adresses de réception proviennent bien d'utilisateurs légitimes, puisqu'il a signé leur version aveuglée auparavant avec sa clé privée. En revanche, il ne peut pas associer une adresse de sortie spécifique à un utilisateur donné en entrée. Il n'y a donc aucun lien entre les entrées et les sorties, même du point de vue du coordinateur. Une fois la transaction construite par le coordinateur, il la renvoie aux participants qui la signent afin de déverrouiller leur input, après avoir vérifié que leur output se trouve bien dans cette transaction. Les participants envoient la signature au coordinateur. Une fois toutes les signatures collectées, le coordinateur peut diffuser la transaction coinjoin sur le réseau Bitcoin.



Cette méthode garantit que le coordinateur ne peut ni compromettre l'anonymat des participants, ni voler les bitcoins durant tout le processus de coinjoin.

Il est difficile de déterminer avec certitude qui a introduit en premier l'idée du coinjoin sur Bitcoin, et qui a eu l'idée d'utiliser les signatures aveugles de David

Chaum dans ce contexte. On pense souvent que c'est Gregory Maxwell qui en a parlé en premier dans un message sur BitcoinTalk en 2013 :

> *« En utilisant les signatures aveugles de Chaum : Les utilisateurs se connectent et fournissent des inputs (et des adresses de change) ainsi qu'une version cryptographiquement aveuglée de l'adresse vers laquelle ils souhaitent envoyer leurs pièces privées ; le serveur signe les jetons et les leur renvoie. Les utilisateurs se reconnectent anonymement, démasquent leurs adresses de sortie et les renvoient au serveur. Le serveur peut voir que toutes les sorties ont été signées par lui et que, par conséquent, toutes les sorties proviennent de participants valides. Plus tard, les personnes se reconnectent et signent. »*

Maxwell, G. (2013, 22 août). *CoinJoin : Bitcoin privacy for the real world*. BitcoinTalk Forum. https://bitcointalk.org/index.php?topic=279249.0

Toutefois, il existe d'autres mentions antérieures, à la fois pour les signatures de Chaum dans le cadre du mixage, mais également pour les coinjoins. En juin 2011, Duncan Townsend présente sur BitcoinTalk un mélangeur qui utilise les signatures de Chaum d'une manière assez similaire aux coinjoins chaumiens modernes. Dans le même thread, on peut retrouver un message de hashcoin en réponse à Duncan Townsend pour améliorer son mélangeur. Ce message présente justement ce qui ressemble le plus aux coinjoins. On retrouve également une mention d'un système similaire dans un message d'Alex Mizrahi en 2012, alors qu'il conseillait les créateurs de Tenebrix. Le terme en lui-même de « coinjoin » n'aurait pas été inventé par Greg Maxwell, mais il viendrait d'une idée de Peter Todd.

Terme associé : **COINJOIN**

CHECKPOINTS

✪ PROTOCOLE | FR : POINTS DE CONTROLE

Empreintes de blocs spécifiques qui étaient codées en dur dans Bitcoin Core. Chaque checkpoint associait une hauteur de bloc à un hash attendu, et tout fork en dessous d'un checkpoint déjà synchronisé était rejeté.

Introduit par Satoshi Nakamoto le 17 juillet 2010 (version 0.3.2), ce mécanisme visait à empêcher toute réécriture de la chaîne historique, même par un attaquant disposant d'une majorité de la puissance de calcul. En pratique, les checkpoints servaient surtout de protection anti-DoS lors de l'IBD : sans eux, un attaquant pouvait exploiter le bug du *time warp* pour submerger la mémoire d'un nouveau nœud avec des millions d'en-têtes à difficulté minimale. Mais ces checkpoints n'ajoutaient pas de réelle sécurité au consensus : si les checkpoints devaient influencer le choix de la chaîne, le système serait déjà compromis. Le dernier a été ajouté au bloc 295 000, en avril 2014.

Les checkpoints ont été retirés dans Bitcoin Core 30.0 (octobre 2025), remplacés par le mécanisme de *headers presync* (août 2022) qui vérifie la preuve de travail d'une chaîne avant de stocker ses en-têtes, éliminant toute donnée

codée en dur par les développeurs.

Termes associés : ASSUME VALID | BITCOIN CORE | HEADERS-FIRST SYNC | IBD - INITIAL BLOCK DOWNLOAD

CHEMIN DE DÉRIVATION

PORTEFEUILLE | EN : DERIVATION PATH

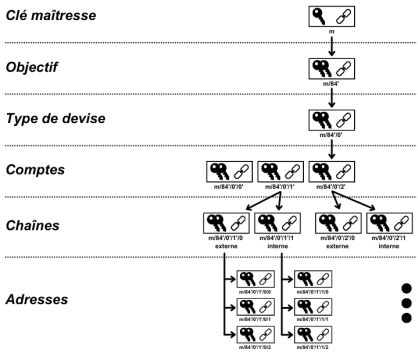
Dans le cadre des portefeuilles déterministes et hiérarchiques (HD), un chemin de dérivation désigne la séquence d'index utilisée pour dériver des clés enfants à partir d'une clé maîtresse. Décrit dans le BIP-0032, ce chemin permet d'identifier la structure arborescente de dérivation des clés enfants. Un chemin de dérivation est représenté par une série d'index séparés par des barres obliques, et commence toujours par la clé maîtresse (notée `m/`). Par exemple, un chemin typique pourrait être `m/84'/0'/0'/0/0`. Chaque niveau de dérivation est associé à une profondeur spécifique :

- `m /` indique la clé privée maîtresse. Elle est unique pour un portefeuille et ne peut pas avoir de sœurs à la même profondeur. La clé maîtresse est dérivée directement depuis la graine ;
- `m / purpose' /` indique l'objectif de dérivation qui permet d'identifier le standard suivi. Ce champ est décrit dans le BIP-0043. Par exemple, si le portefeuille respecte le standard BIP-0084 (SegWit V0), l'index sera alors `84'` ;
- `m / purpose' / coin-type' /` indique le type de cryptomonnaie. Cela permet de bien différencier les branches dédiées à une cryptomonnaie, des branches dédiées à une autre cryptomonnaie sur un portefeuille multi-coin. L'index dédié au bitcoin est le `0'` ;
- `m / purpose' / coin-type' / account' /` indique le numéro de compte. Cette profondeur permet de différencier et d'organiser facilement un portefeuille en différents comptes. Ces comptes sont numérotés à partir de `0'`. Les clés étendues (`xpub`, `xprv`...) se trouvent à ce niveau de profondeur ;
- `m / purpose' / coin-type' / account' / change /` indique la chaîne. Chaque compte tel que défini en profondeur 3 dispose de deux chaînes en profondeur 4 : une chaîne externe et une chaîne interne (également appelée « change »). La chaîne externe dérive des adresses destinées à être communiquées publiquement, c'est-à-dire les adresses que l'on nous propose lorsque l'on clique sur « recevoir » dans notre logiciel de portefeuille. La chaîne interne dérive les adresses destinées à ne pas être échangées publiquement, c'est-à-dire principalement les adresses de change. La chaîne externe est identifiée avec l'index `0` et la chaîne interne est identifiée avec l'index `1`. Vous remarquerez qu'à partir de cette profondeur, on ne réalise plus une dérivation endurcie, mais une dérivation normale (il n'y a pas d'apostrophe). C'est grâce à ce mécanisme que l'on est capable de dériver l'ensemble des clés publiques enfants à partir de leur `xpub` ;
- `m / purpose' / coin-type' / account' / change / address-index` indique simplement le numéro de l'adresse de réception et de sa paire de clés, afin de la différencier de ses sœurs à la même profondeur sur la même branche. Par exemple, la première adresse dérivée dispose de l'index `0`, la

deuxième adresse dispose de l'index `1`, etc.

Par exemple, si mon adresse de réception dispose du chemin de dérivation `m / 86' / 0' / 0' / 0 / 5`, on peut en déduire les informations suivantes :

- `86'` indique que nous suivons le standard de dérivation du BIP-0086 (Taproot / SegWit V1) ;
- `0'` indique que c'est une adresse Bitcoin ;
- `0'` indique que l'on est sur le premier compte du portefeuille ;
- `0` indique que c'est une adresse externe ;
- `5` indique que c'est la sixième adresse externe de ce compte.



Termes associés : **BIP-0032** **XPUB**

CHEMIN DE RÉCUPÉRATION

PORTEFEUILLE | EN : RECOVERY PATH

Dans un logiciel de portefeuille utilisant Miniscript, comme Liana par exemple, le ou les chemins de récupération désignent des ensembles de clés qui ne deviennent opérationnels qu'après une période d'inactivité définie dans le script qui bloque les bitcoins (*timelock*). Les chemins de récupération sont utilisables seulement une fois que les *timelocks* sont passés, et offrent ainsi une méthode de récupération des fonds lorsque le chemin primaire n'est pas disponible. Prenons l'exemple d'un script qui intègre 2 clés distinctes : la clé A, qui autorise la dépense immédiate des bitcoins, et la clé B, qui permet de les dépenser après un délai défini par un *timelock* de 52 560 blocs. Dans cet exemple, la clé A provient du chemin primaire, tandis que la clé B provient du chemin de récupération.

Terme associé : **MINISCRIPT**

CHEMIN PRIMAIRE

PORTEFEUILLE | EN : PRIMARY PATH

Dans un logiciel de portefeuille utilisant Miniscript, comme Liana par exemple, le chemin primaire fait référence à l'ensemble de clés permettant des dépenses immédiates des fonds, sans aucune condition temporelle. Ce chemin est toujours accessible et est utilisé pour la gestion quotidienne des bitcoins, sans

nécessiter d'attente (*timelock*) contrairement aux chemins de récupération. Prenons l'exemple d'un script qui intègre 2 clés distinctes : la clé A, qui autorise la dépense immédiate des bitcoins, et la clé B, qui permet de les dépenser après un délai défini par un *timelock* de 52 560 blocs. Dans cet exemple, la clé A provient du chemin primaire, tandis que la clé B provient du chemin de récupération.

Terme associé : **MINISCRIPT**

CHIFFRER - CHIFFREMENT

CRYPTOGRAPHIE | EN : ENCRYPT - ENCRYPTION

Méthode cryptographique permettant de convertir une information brute en information chiffrée. Une information chiffrée masque la signification originale des données pour empêcher qu'elles ne soient connues. Le chiffrement consiste en une série de transformations effectuées sur l'information originale à l'aide d'une clé. Si ces transformations sont réversibles, le processus d'inversion correspondant est appelé « déchiffrement », et il permet de restaurer les informations à leur état brut.

Termes associés : **CONDENSAT** **FONCTION DE HACHAGE**

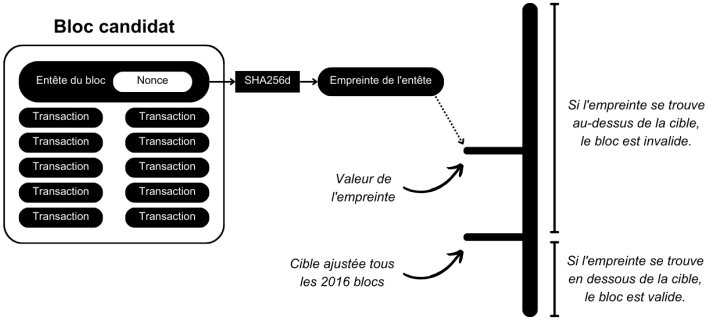
CIBLE DE DIFFICULTÉ

MINAGE | EN : DIFFICULTY TARGET

Le facteur de difficulté, aussi connu sous le nom de cible de difficulté, est un paramètre utilisé dans le mécanisme de consensus par preuve de travail (*Proof of Work*, PoW) sur Bitcoin. La cible représente une valeur numérique qui détermine la difficulté pour les mineurs de résoudre un problème cryptographique spécifique, appelé preuve de travail, lors de la création d'un nouveau bloc sur la blockchain.

La cible de difficulté est un nombre ajustable de 256 bits (32 octets) déterminant une limite d'acceptabilité pour le hachage de l'entête des blocs. Autrement dit, pour qu'un bloc soit valide, le hachage de son entête avec `SHA256d` (double `SHA256`) doit être numériquement inférieur ou égal à la cible de difficulté. La preuve de travail consiste à modifier le champ `nonce` de l'entête du bloc ou de la transaction coinbase jusqu'à ce que le hachage résultant soit inférieur à la valeur cible.

Cette cible est ajustée tous les 2016 blocs (environ toutes les deux semaines), lors d'un évènement que l'on appelle « ajustement ». Le facteur de difficulté est recalculé en fonction du temps qu'il a fallu pour miner les 2016 blocs précédents. Si le temps total est inférieur à deux semaines, la difficulté augmente en ajustant la cible à la baisse. Si le temps total est supérieur à deux semaines, la difficulté diminue en ajustant la cible à la hausse. L'objectif est de conserver un temps de minage par bloc moyen à 10 minutes. Ce temps entre chaque bloc permet d'éviter les divisions du réseau Bitcoin, résultant en un gaspillage de la puissance de calcul. La cible de difficulté se trouve dans chaque entête de bloc, au sein du champ `nBits`. Puisque ce champ est réduit à 32 bits et que la cible fait en réalité 256 bits, la cible est compactée dans un format scientifique moins précis.



La cible de difficulté est parfois également nommée « facteur de difficulté ». Par extension, on peut l'évoquer avec son encodage dans les entêtes de bloc avec le terme « nBits ».

Termes associés : **AJUSTEMENT DE LA DIFFICULTE** | **PREUVE DE TRAVAIL**

CIOH

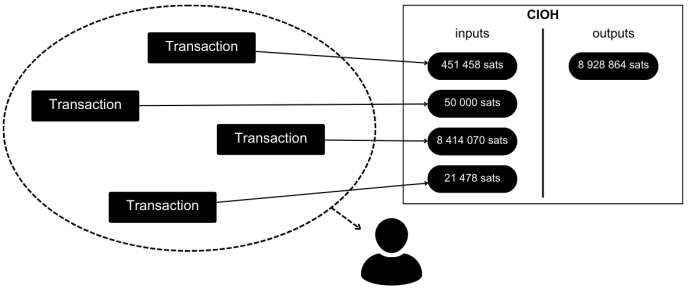
CONFIDENTIALITÉ | FR : HEURISTIQUE DE PROPRIÉTÉ COMMUNE DES ENTRÉES

Sigle de « *Common Input Ownership Heuristic* ». C'est une heuristique utilisée dans le domaine de l'analyse de chaîne sur Bitcoin qui suppose que toutes les entrées d'une transaction appartiennent à une même entité ou à un même utilisateur. Lorsque l'on observe les données publiques d'une transaction Bitcoin, et que l'on y repère plusieurs entrées (inputs), alors, s'il n'y a pas de *patterns* ou d'autres informations qui viendraient infirmer cela, on peut estimer que toutes les entrées de cette transaction appartenaient à une seule et même personne (ou entité).

Cette heuristique d'analyse a été découverte par Satoshi Nakamoto lui-même, qui en parle dans la partie 10 du White Paper :

> « *Toutefois, la liaison est inévitable avec les transactions multi-entrées, qui révèlent nécessairement que leurs entrées étaient détenues par un même propriétaire. Le risque est que si le propriétaire d'une clef est révélé, les liaisons peuvent révéler d'autres transactions qui ont appartenu au même propriétaire.* » - Nakamoto, S. (2008). « Bitcoin : A Peer-to-Peer Electronic Cash System ». Consulté à l'adresse https://bitcoin.org/bitcoin.pdf.

Encore aujourd'hui, le CIOH demeure la principale heuristique employée par les sociétés d'analyse de chaîne, avec la réutilisation d'adresse.



Termes associés : **ANALYSE DE CHAINE** | **RÉUTILISATION D'ADRESSE**

CIRCULAR REBALANCE

LIGHTNING NETWORK | FR : RÉÉQUILIBRAGE CIRCULAIRE

Paiement qu'un nœud Lightning s'envoie à lui-même en empruntant un chemin passant par d'autres nœuds du réseau. L'objectif est de déplacer de la liquidité d'un canal vers un autre sans ouvrir ni fermer de canal. Par exemple, si un nœud dispose de beaucoup de liquidité sortante sur un canal A mais manque de liquidité sortante sur un canal B, il peut s'envoyer un paiement circulaire qui sort par A et revient par B, rééquilibrant ainsi les deux canaux. Cette opération a un coût correspondant aux frais de routage prélevés par les nœuds intermédiaires.

Termes associés : **BOS - BALANCE OF SATOSHIS** | **FEE RATE - LIGHTNING**

CISA

PROTOCOLE

Acronyme de « *Cross-Input Signature Aggregation* ». C'est une proposition technique qui a pour objectif d'optimiser la taille des transactions Bitcoin en réduisant le nombre de signatures requises pour les valider.

Actuellement, sur Bitcoin, chaque input d'une transaction doit avoir une signature individuelle pour pouvoir être consommé. Cela permet de prouver que le propriétaire de l'UTXO en question autorise la transaction. Avec CISA, l'objectif est de combiner les signatures de tous les inputs d'une même transaction en une seule signature qui couvre l'ensemble des inputs. Cela permettrait de réduire la taille des transactions qui disposent de nombreux inputs, et donc d'en réduire également les frais. Ce serait notamment très utile pour les personnes qui réalisent des coinjoins ou des consolidations, tout en permettant de confirmer plus de transactions sur Bitcoin sans pour autant modifier la taille ou l'intervalle des blocs. CISA repose sur le protocole de Schnorr qui permet l'agrégation linéaire des signatures. Cela signifie qu'une seule signature peut prouver la possession de plusieurs clés indépendantes.

Cependant, l'implémentation de CISA sur Bitcoin est très complexe, car elle nécessite des modifications profondes dans la manière de fonctionner des scripts. Actuellement, la vérification des scripts sur Bitcoin se fait input par input. Passer à un modèle où une transaction entière est vérifiée d'un coup, comme le

propose CISA, est loin d'être une modification anodine.

Termes associés : **COINJOIN** | **SCHNORR**

CITREA

COUCHE SUPÉRIEURE

Premier rollup à preuves à divulgation nulle de connaissance (ZK rollup) déployé sur Bitcoin, développé par Chainway Labs. Citrea utilise un zkEVM de Type 2, construit avec RISC Zero, qui offre une compatibilité complète avec la machine virtuelle d'Ethereum et permet aux développeurs Solidity de déployer directement leurs contrats sur Bitcoin. Le système traite les transactions hors chaîne par lots, génère des preuves STARK (enveloppées ensuite en preuves SNARK) et publie ces preuves directement sur Bitcoin, qui sert à la fois de couche de disponibilité des données et de couche de règlement. Le pont bidirectionnel de Citrea, nommé Clementine, repose sur BitVM2 et fonctionne avec une hypothèse de confiance 1-sur-N : tant qu'au moins un vérificateur du pont est honnête, les fonds sont en sécurité. Le bitcoin natif sur Citrea est désigné cBTC. La plateforme propose également ctUSD, un stablecoin adossé au dollar et garanti par des bons du Trésor américain, émis en partenariat avec MoonPay et M0. Citrea ne nécessite aucune modification des règles de consensus de Bitcoin. Le *mainnet* est en ligne depuis janvier 2026.

Termes associés : **CLEMENTINE** | **RISC ZERO** | **ROLLUP**

CLARK - COVENANTLESS ARK

COUCHE SUPÉRIEURE

Version du protocole Ark implémentable sans nécessiter l'ajout de covenants à Bitcoin via un soft fork. Le nom « clArk » est la contraction de « *covenantless Ark* ». Lors de la proposition initiale d'Ark en mai 2023 par Burak Keceli, le protocole requérait des covenants, pas encore disponibles sur Bitcoin. La version clArk, développée à partir de 2024, contourne cette limitation.

En l'absence de covenants, clArk émule les conditions de dépense en demandant à tous les participants d'un round de signer des transactions multisig n-parmi-n. Chaque participant doit signer l'intégralité de l'arbre VTXO composant le round, là où les covenants permettraient de ne signer que les clauses concernant ses propres fonds. Cette approche entraîne une surcharge significative en bande passante, en stockage et en interactivité : tous les participants doivent être en ligne simultanément durant le round.

L'ajout éventuel de covenants à Bitcoin améliorerait substantiellement le protocole : les utilisateurs n'auraient plus besoin d'être connectés en permanence, la gestion de liquidité des ASP serait optimisée, et le protocole pourrait mieux gérer certains scénarios. ClArk représente donc un compromis permettant de déployer Ark sur Bitcoin tel qu'il existe aujourd'hui, sans avoir besoin de faire un soft fork.

Termes associés : **ARK** | **VTXO**

CLÉ DE RÉVOCATION

LIGHTNING NETWORK | EN : REVOCATION KEY

Clé cryptographique utilisée dans le Lightning Network pour permettre la punition d'un pair qui publie un ancien état de canal. Chaque transaction d'engagement contient une sortie `to_local` qui offre deux chemins de dépense :

- l'un soumis à un délai relatif (`OP_CHECKSEQUENCEVERIFY`) pour le propriétaire,
- l'autre immédiatement dépensable par la contrepartie si elle détient la clé de révocation correspondante.

La clé de révocation est construite de manière collaborative. Pour une transaction d'engagement donnée, Alice fournit un point d'engagement (C_A) et Bob fournit un point de base de révocation (R_B). La clé publique de révocation est calculée comme suit :

$$Rev \;=\; R_B \cdot SHA256(R_B \parallel C_A) \;+\; C_A \cdot SHA256(C_A \parallel R_B)$$

Ni Alice ni Bob ne peuvent dériver seuls la clé privée correspondante. Lorsque l'état du canal est mis à jour, Alice révèle le secret c_A associé à son point d'engagement. Bob peut alors combiner c_A avec son propre secret r_B pour obtenir la clé privée de révocation et, le cas échéant, construire une transaction de pénalité.

Terme associé : **TRANSACTION DE PÉNALITÉ**

CLÉ ÉTENDUE

PORTEFEUILLE | EN : EXTENDED KEY

Suite de caractères qui combine une clé (publique ou privée), son code de chaîne associé et une série de métadonnées. Une clé étendue rassemble en un seul identifiant tous les éléments nécessaires à la dérivation de clés enfants. Elles sont utilisées dans les portefeuilles déterministes et hiérarchiques, et peuvent être de deux types : une clé publique étendue (utilisée pour dériver des clés publiques enfants) ou une clé privée étendue (utilisée pour dériver à la fois des clés privées et des clés publiques enfants). Une clé étendue inclut donc plusieurs données différentes, décrites au sein du BIP-0032, dans l'ordre :

- Le préfixe : `prv` et `pub` sont des HRP permettant d'indiquer si l'on a affaire à une clé privée étendue (`prv`) ou à une clé publique étendue (`pub`). La première lettre du préfixe permet, elle, de désigner la version de la clé étendue : `x` pour Legacy ou SegWit V1 sur Bitcoin, `t` pour Legacy ou SegWit V1 sur Bitcoin Testnet, `y` pour Nested SegWit sur Bitcoin, `u` pour Nested SegWit sur Bitcoin Testnet, `z` pour SegWit V0 sur Bitcoin, `v` pour SegWit V0 sur Bitcoin Testnet.
- La profondeur, qui indique le nombre de dérivations intervenues depuis la clé maîtresse pour arriver jusqu'à la clé étendue ;
- L'empreinte du parent. Cela représente les 4 premiers octets du `HASH160` de la clé publique parent ;
- L'index. C'est le numéro de la paire parmi ses sœurs dont est issue la clé étendue ;

- Le code de chaîne ;
- Un octet de rembourrage si c'est une clé privée `0×00` ;
- La clé privée ou la clé publique ;
- Une somme de contrôle. Elle représente les 4 premiers octets du `HASH256` de tout le reste de la clé étendue.

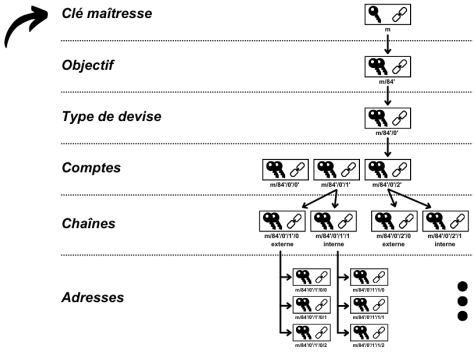
Dans la pratique, la clé publique étendue est utilisée pour générer des adresses de réception et pour observer les transactions d'un compte, sans exposer les clés privées associées. Cela peut permettre, par exemple, la création d'un portefeuille dit « watch-only ». Il est toutefois important de noter que la clé publique étendue est une information sensible pour la confidentialité de l'utilisateur, car sa divulgation peut permettre à des tiers de tracer les transactions et de voir le solde du compte associé.

Termes associés : **BIP-0032** **CODE DE CHAINE**

CLÉ MAITRESSE

PORTEFEUILLE | EN : MASTER KEY

Dans le cadre des portefeuilles HD (déterministes et hiérarchiques) la clé privée maîtresse est une clé privée unique dérivée depuis la graine (seed) du portefeuille. Pour obtenir la clé maîtresse, on applique la fonction `HMAC-SHA512` à la graine, en utilisant littéralement les mots « *Bitcoin seed* » comme clé. Le résultat de cette opération donne un output de 512 bits, dont les 256 premiers bits constituent la clé maîtresse, et les 256 bits restants forment le code de chaîne maître. La clé maîtresse et le code de chaîne maître servent de point de départ pour dériver toutes les clés privées et publiques enfants dans l'arborescence du portefeuille HD. La clé privée maîtresse est donc en profondeur 0 de dérivation.



Termes associés : **DÉRIVATION** **GRAINE**

CLÉ PRIVÉE

PORTEFEUILLE | EN : PRIVATE KEY

Une clé privée est un élément fondamental de la cryptographie asymétrique. Il s'agit d'un nombre (de 256 bits dans le cadre de Bitcoin) qui représente un secret cryptographique. Cette clé est utilisée pour signer numériquement des transactions et prouver la possession d'une clé publique Bitcoin (et par extension, d'une adresse de réception) en satisfaisant un `scriptPubKey`. Les clés privées permettent donc de dépenser des bitcoins en débloquant les UTXO associés à la clé publique correspondante. Les clés privées doivent être conservées strictement confidentielles, car leur divulgation pourrait permettre à des tiers malveillants de prendre le contrôle des fonds associés.

Dans le système Bitcoin, la clé privée est liée à une clé publique par le biais d'un algorithme de signature numérique à courbes elliptiques (ECDSA ou Schnorr). La clé publique est dérivée de la clé privée, mais l'inverse est pratiquement impossible à réaliser en raison de la difficulté computationnelle inhérente à la résolution du problème mathématique sous-jacent (problème du logarithme discret). La clé publique est généralement utilisée pour générer une adresse Bitcoin, qui sert à bloquer des bitcoins à l'aide d'un script. En cryptographie, les clés privées sont souvent des nombres aléatoires ou pseudo-aléatoires. Dans le contexte spécifique des portefeuilles déterministes et hiérarchiques Bitcoin, les clés privées sont dérivées de manière déterministe depuis la graine (seed). Les clés privées sont fréquemment confondues avec la graine (seed) ou avec la phrase de récupération (mnémonique). Pourtant, ces éléments sont bien différents.

En anglais, une clé privée se dit « private key ». Ce terme est parfois abrégé avec « privkey », ou « PV ».

Termes associés : **CLÉ PUBLIQUE** | **ECDSA**

CLÉ PUBLIQUE

PORTEFEUILLE | EN : PUBLIC KEY

La clé publique est un élément utilisé dans la cryptographie asymétrique. Elle est générée à partir d'une clé privée en utilisant une fonction mathématique irréversible. Sur Bitcoin, les clés publiques sont dérivées depuis leur clé privée associée grâce aux algorithmes de signature numérique à courbes elliptiques ECDSA ou Schnorr. La clé publique, contrairement à la clé privée, peut être partagée librement sans compromettre la sécurité des fonds. Dans le cadre du protocole Bitcoin, la clé publique sert de base pour créer une adresse Bitcoin, qui est ensuite utilisée pour créer des conditions de dépense sur un UTXO à l'aide d'un `scriptPubKey`. Les clés publiques sont fréquemment confondues avec la clé maîtresse ou avec les clés étendues (`xpub`...). Pourtant, ces éléments sont bien différents.

En anglais, une clé publique se dit « public key ». Ce terme est parfois abrégé avec « pubkey », ou « PK ».

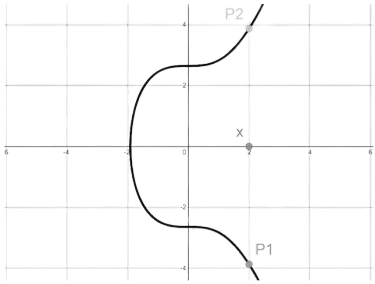
Termes associés : **ADRESSE DE RÉCEPTION** | **CLÉ PRIVÉE**

CLÉ PUBLIQUE COMPRESSÉE

PORTEFEUILLE	EN : COMPRESSED PUBLIC KEY

Une clé publique est utilisée dans les scripts (soit directement sous la forme d'une clé publique, soit sous la forme d'une adresse) pour recevoir et sécuriser des bitcoins. Une clé publique brute est représentée par un point sur une courbe elliptique, composé de deux coordonnées (`x, y`) chacune de 256 bits. En format brut, une clé publique mesure donc 512 bits, sans compter l'octet supplémentaire pour identifier le format. Une clé publique compressée, en revanche, est une forme plus compacte de représentation d'une clé publique. Elle utilise seulement la coordonnée `x` et un préfixe (`02` ou `03`) qui indique la parité de la coordonnée `y` (pair ou impair).

Si l'on simplifie cela au corps des réels, la courbe elliptique étant symétrique par rapport à l'axe des abscisses, pour tout point P (`x, y`) sur la courbe, il existe un point P' (`x, -y`) qui sera également sur cette même courbe. Cela signifie qu'à chaque `x` correspondent seulement deux valeurs possibles de `y`, positive et négative. Par exemple, pour une abscisse `x` donnée, il y aurait deux points P_1 et P_2 sur la courbe elliptique, qui partagent la même abscisse, mais avec des ordonnées opposées :



Pour choisir entre les deux points potentiels sur la courbe, on ajoute à `x` un préfixe spécifiant quel `y` choisir. Cette méthode permet de réduire la taille d'une clé publique de 520 bits à seulement 264 bits (8 bits de préfixe + 256 bits pour `x`). Cette représentation est connue sous le nom de forme compressée de la clé publique.

Cependant, dans le cadre de la cryptographie sur les courbes elliptiques, nous utilisons non pas les réels, mais un corps fini d'ordre `p` (un nombre premier). Dans ce contexte, le « signe » de `y` est déterminé par sa parité, c'est-à-dire si `y` est pair ou impair. Le préfixe `0×02` indique alors un `y` pair, tandis que `0×03` indique un `y` impair.

Considérons l'exemple suivant d'une clé publique brute (un point sur la courbe elliptique) en hexadécimal :

```
K=04678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61
deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6bf1
1d5f
```

On peut isoler le préfixe, `x`, et `y` :

```
Préfixe=04
x=678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb6
y=49f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5f
```

Pour déterminer la parité de `y`, on examine le dernier caractère hexadécimal de `y` :

```
Base 16 (hexadécimal) : f
Base 10 (décimal) : 15
y est impair
```

Le préfixe pour la clé publique compressée sera `03`. La clé publique compressée en hexadécimal devient :

```
K=03678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb6
```

Termes associés : **CLÉ PUBLIQUE** | **COURBE ELLIPTIQUE**

CLEARNET

RÉSEAU

Terme désignant l'internet conventionnel, par opposition aux réseaux superposés comme Tor ou I2P. Sur le *clearnet*, les communications transitent avec des adresses IP identifiables, ce qui les rend observables par les fournisseurs d'accès, les opérateurs réseau ou tout intermédiaire.

Dans le contexte de Bitcoin et du Lightning Network, utiliser le *clearnet* signifie que l'adresse IP d'un nœud est directement visible par ses pairs et par les observateurs du réseau. De nombreux opérateurs de nœuds préfèrent se connecter exclusivement via Tor pour protéger leur identité, même si le *clearnet* offre des connexions plus rapides et plus stables. Certaines configurations hybrides permettent aux nœuds d'être accessibles à la fois sur Tor et sur le *clearnet*.

Terme associé : **TOR**

CLEMENTINE

COUCHE SUPÉRIEURE

Pont bidirectionnel à confiance minimisée développé par Chainway Labs pour le rollup Citrea. Clementine permet de transférer des BTC entre la chaîne principale Bitcoin et Citrea (BTC ↔ cBTC) sans recourir à un multisig de confiance classique. Le protocole repose sur BitVM2 et fonctionne avec une hypothèse de confiance 1-sur-N : il suffit qu'un seul participant du groupe de vérificateurs soit honnête pour que les fonds soient protégés contre le vol. Pour un dépôt (*peg-in*), l'utilisateur envoie des BTC vers une adresse Taproot encodant deux chemins de dépense : un chemin permet aux signataires de déplacer les fonds vers un coffre (*vault*), l'autre permet un remboursement automatique après un *timelock* si le pont ne réagit pas. Pour un retrait (*peg-out*), l'utilisateur brûle ses

cBTC sur Citrea, puis un opérateur avance les fonds en BTC sur Bitcoin et se fait ensuite rembourser depuis le coffre, sous réserve qu'aucun contestataire ne prouve un comportement frauduleux via un mécanisme de contestation utilisant BitVM2. Clementine est open source.

Termes associés : **BITVM** **CITREA**

CLI

‹/› INFORMATIQUE

Acronyme de « *Command Line Interface* », ou « interface en ligne de commande » en français. C'est une méthode d'interaction avec des logiciels qui repose sur la saisie de commandes textuelles dans un terminal ou une console. La CLI se différencie de la GUI (interface graphique utilisateur) qui dispose de méthodes d'interactions de pointage (avec la souris) et d'éléments visuels interactifs.

Terme associé : **GUI**

CLIENT-SIDE VALIDATION

RGB | FR : VALIDATION CÔTÉ CLIENT

Processus par lequel chaque partie (client) vérifie un ensemble de données échangées en privé, selon les règles d'un protocole. Dans le cas du protocole RGB, ces données échangées sont regroupées dans ce qu'on appelle des consignments. Contrairement au protocole Bitcoin qui exige que toutes les transactions soient publiées on-chain, RGB permet de ne stocker en public que des commitments (ancrés dans Bitcoin), tandis que l'essentiel des informations de contrat (transitions, attestations, preuves) reste off-chain, partagé seulement entre les utilisateurs concernés.

Termes associés : **CONSIGNMENT** **RGB**

CLN - C-LIGHTNING

LIGHTNING NETWORK

Ancien nom de l'implémentation Core-Lightning.

Terme associé : **CLN - CORE-LIGHTNING**

CLN - CORE-LIGHTNING

LIGHTNING NETWORK

Implémentation majeure du protocole Lightning Network écrite en langage C et Rust. Développée par Blockstream, Core-Lightning est conçue pour être légère et performante. Elle se distingue par son architecture modulaire, permettant aux développeurs d'ajouter facilement des fonctionnalités personnalisées. Cette implémentation a été renommée en 2022. Son nom original était auparavant « C-Lightning ».

Termes associés : **ECLAIR** **LND**

CLONE

</> INFORMATIQUE

Dans le cadre de Git, consiste à créer une copie locale d'un dépôt distant. Cette opération télécharge l'ensemble du dépôt, y compris toutes les branches et l'historique des commits. En tant qu'utilisateur de Bitcoin, il est possible d'avoir à utiliser cette commande lorsque l'on télécharge un logiciel.

Termes associés : **FORK - GIT** | **GIT**

CLUSTER

CONFIDENTIALITÉ

Dans le cadre de l'analyse de chaîne, un cluster est un ensemble d'adresses de réception qui sont associées à une même entité par un analyste. En utilisant diverses heuristiques, il est possible de détecter des activités *on-chain* qui semblent émaner d'une unique personne ou organisation. Ce regroupement d'activités forme ce que l'on appelle un *cluster*. L'objectif de l'analyse de chaîne est souvent d'identifier un point d'entrée dans ce cluster, permettant ainsi de lier toutes ces activités à une forme d'identité dans le monde réel.

Termes associés : **ANALYSE DE CHAINE** | **HEURISTIQUE D'ANALYSE**

CLUSTER MEMPOOL

PROTOCOLE

Proposition de refonte de la gestion de la mempool dans Bitcoin Core. La cluster mempool organise les transactions non confirmées en « grappes » (*clusters*) : des ensembles de transactions liées par des relations parent-enfant. Au sein de chaque grappe, un algorithme de linéarisation pré-calcule l'ordre optimal de minage et regroupe les transactions en « morceaux » (*chunks*) triés par taux de frais décroissant.

Cette refonte corrige une incohérence dans le fonctionnement actuel de la mempool. La sélection des transactions pour le minage utilise le taux de frais ancêtre le plus élevé, tandis que l'éviction utilise le taux de frais descendant le plus faible. Comme ces deux heuristiques ne sont pas symétriques, il peut arriver que la meilleure transaction de la mempool soit évincée en premier. Avec la cluster mempool, la construction de blocs et l'éviction deviennent des opérations exactement inverses : on inclut les morceaux du plus rentable au moins rentable, et on évicte dans l'ordre inverse.

Cette approche est viable grâce à la limitation de la taille des grappes, qui remplace les anciennes limites d'ancêtres et de descendants. Comme les morceaux optimaux ne franchissent jamais les frontières entre grappes, l'ajout d'une transaction ne nécessite de recalculer que sa propre grappe. Le taux de frais de chaque morceau sert alors de score unifié pour toutes les opérations de la mempool : construction de blocs, éviction, évaluation du RBF, relais de transactions et estimation des frais.

Terme associé : **CPFP - CHILD PAY FOR PARENT**

CODE DE CHAINE

PORTEFEUILLE | EN : CHAIN CODE

Dans le contexte de la dérivation hiérarchique et déterministe (HD) des portefeuilles Bitcoin, le code de chaîne est une valeur de sel cryptographique de 256 bits utilisée pour générer des clés enfants à partir d'une clé parent, selon le standard BIP-0032. Le code de chaîne est utilisé en combinaison avec la clé parent et l'index de l'enfant pour générer de manière déterministe une nouvelle paire de clés (clé privée et clé publique) sans révéler la clé parent ou les autres clés enfants dérivées.

Il existe donc un code de chaîne unique pour chaque paire de clés. Le code de chaîne est obtenu soit en hachant la graine du portefeuille, et en prenant la moitié des bits à droite. Dans ce cas, on parle d'un code de chaîne maître, associé à la clé privée maîtresse. Ou bien, il peut être obtenu en hachant une clé parent avec son code de chaîne parent et un index, puis en conservant les bits de droite. On parle alors de code de chaîne enfant.

Il est impossible de dériver des clés sans avoir la connaissance du code de chaîne associé à chaque paire parent. Il permet d'introduire des données pseudo-aléatoires dans le processus de dérivation pour garantir que la génération des clés cryptographiques reste imprévisible pour les attaquants tout en étant déterministe pour le détenteur du portefeuille.

Termes associés : **BIP-0032** **CLÉ ÉTENDUE**

CODE DE CHAINE MAITRE

PORTEFEUILLE | EN : MASTER CHAIN CODE

Désigne le code de chaîne associé à la clé maîtresse du portefeuille, à la base de l'arbre de dérivation de toutes les clés.

Terme associé : **CODE DE CHAINE**

CODE DE PAIMENT RÉUTILISABLE

CONFIDENTIALITÉ | EN : REUSABLE PAYMENT CODE

Dans le BIP-0047, un code de paiement réutilisable est un identifiant statique généré à partir d'un portefeuille Bitcoin permettant d'engager une transaction de notification et de dériver des adresses uniques. Cela permet de ne pas faire de réutilisation d'adresses, qui mènent à une perte de la confidentialité, sans pour autant devoir dériver et transmettre manuellement de nouvelles adresses vierges à chaque paiement. Dans le BIP-0047, les codes de paiement réutilisables sont construits de la manière suivante :

- L'octet 0 correspond à la version ;
- L'octet 1 est un champ de bits permettant d'ajouter des informations en cas d'utilisation spécifique ;
- L'octet 2 permet d'indiquer la parité du `y` de la clé publique ;
- De l'octet 3 à l'octet 34, on retrouvera la valeur `x` de la clé publique ;
- De l'octet 35 à l'octet 66, il y a le code de chaîne associé à la clé publique ;
- De l'octet 67 à l'octet 79, il y a du rembourrage de zéros.

On ajoute généralement un octet de version (`0×47`) au départ du code de paie-

ment et une somme de contrôle à la fin, puis on l'encode en Base58Check. La construction d'un code de paiement est donc assez proche de celle d'une clé étendue. Voici mon ancien code de paiement BIP-0047 en Base58Check par exemple :

```
PM8TJSBiQmNQDwTogMAbyqJe2PE2kQXjtgh88MRTxsrnHC8zpEtJ8j7Aj628oUFk
8X6P5rJ7P5qDudE4Hwq9JXSRzGcZJbdJAjM9oVQ1UKU5j2nr7VR5
```

Dans l'implémentation PayNym du BIP-0047, les codes de paiement peuvent également être exprimés sous la forme d'identifiants associés à l'image d'un robot. Voici mon ancien par exemple :

```
+throbbingpond8B1
```

L'utilisation de codes de paiements avec l'implémentation PayNym est actuellement disponible sur Sparrow Wallet sur PC et sur Ashigaru sur mobile (un fork de Samourai Wallet, dont les serveurs ont été saisis par les autorités américaines en avril 2024).

Termes associés : **BIP-0047** **PAYNYM**

CODEX32

PORTEFEUILLE

Schéma d'encodage et de sauvegarde de graines BIP-0032 sur papier, spécifié dans le BIP-0093. Codex32 se distingue des alternatives existantes par la possibilité de vérifier manuellement l'intégrité d'une sauvegarde, sans recourir à aucun appareil électronique.

Le système repose sur le partage de secret de Shamir : la graine peut être fractionnée en plusieurs parts (*shares*), dont seul un sous-ensemble configurable (quorum) est nécessaire pour reconstituer le secret. Chaque part est encodée sous la forme d'une chaîne de caractères Bech32 qui intègre une somme de contrôle (*checksum*). C'est cette somme de contrôle qui rend Codex32 unique : elle peut être calculée et vérifiée entièrement à la main grâce à des tables de correspondance ou à des gabarits rotatifs en papier appelés « volvelles ». L'opération reste laborieuse, mais elle garantit qu'un utilisateur peut auditer ses sauvegardes dans un environnement totalement hors ligne et sans aucun composant numérique.

Codex32 prend en charge les graines de 128 et 256 bits. Il constitue une alternative au schéma SLIP39 de SatoshiLabs, qui offre également le partage de Shamir mais exige un dispositif électronique pour la vérification. Core Lightning a intégré la prise en charge de Codex32 pour la sauvegarde et la restauration de graines en 2023.

Terme associé : **GRAINE**

COIN CONTROL

PORTEFEUILLE | FR : CONTRÔLE DES PIÈCES

Fonctionnalité présente dans certains logiciels de portefeuille Bitcoin, qui donne aux utilisateurs la capacité de sélectionner manuellement les UTXOs spéci-

fiques à utiliser en tant qu'entrées pour effectuer une transaction. En d'autres termes, le coin control offre la possibilité de choisir précisément quels morceaux de bitcoins seront dépensés. Cette fonctionnalité est similaire à l'action de choisir une pièce spécifique de votre porte-monnaie pour payer votre baguette.

Le coin control est particulièrement utile pour gérer ses frais de transaction et pour améliorer sa confidentialité. En effet, en sélectionnant spécifiquement les UTXOs à consommer, les utilisateurs peuvent éviter de fusionner des UTXOs issus de sources différentes, ce qui pourrait révéler des informations sur l'ensemble de leurs fonds (CIOH). Cette fonctionnalité va souvent de pair avec la possibilité d'étiqueter les sorties de transaction.

Termes associés : **CIOH** | **UTXO**

COINBASE

PROTOCOLE

La transaction coinbase est une transaction spéciale et unique incluse dans chaque bloc de la blockchain Bitcoin. Elle représente la première transaction d'un bloc et est créée par le mineur qui a réussi à trouver un en-tête validant la preuve de travail (*Proof-of-Work*), c'est-à-dire inférieur ou égal à la cible.

La transaction coinbase sert principalement deux objectifs : attribuer la récompense de bloc au mineur et ajouter de nouvelles unités de bitcoins à la masse monétaire en circulation. La récompense de bloc, qui est l'incitation économique pour les mineurs à contribuer à s'adonner à la preuve de travail, comprend les frais accumulés pour les transactions incluses dans le bloc et un montant déterminé de bitcoins nouvellement créés ex-nihilo (subvention de bloc). Ce montant, initialement fixé à 50 bitcoins par bloc en 2009, est réduit de moitié tous les 210 000 blocs (environ tous les 4 ans) lors d'un événement appelé « halving ».

La transaction coinbase diffère des transactions régulières de plusieurs manières. Tout d'abord, elle n'a pas d'entrée (input), ce qui signifie qu'aucune sortie de transaction existante (UTXO) n'y est consommée. Ensuite, le script de signature (`scriptSig`) pour la transaction coinbase contient généralement un champ arbitraire permettant d'incorporer des données supplémentaires, telles que des messages personnalisés ou des informations de version de logiciel de minage. Enfin, les bitcoins générés par la transaction coinbase sont soumis à une période de maturité de 100 blocs (101 confirmations) avant de pouvoir être dépensés, afin de prévenir les dépenses potentielles de bitcoins non existants en cas de réorganisation de la chaîne.

Il n'existe aucune traduction de « Coinbase » en français. Il est donc admis d'utiliser directement ce terme.

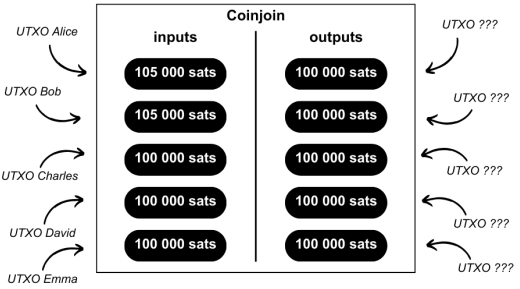
Terme associé : **PÉRIODE DE MATURITÉ**

COINJOIN

CONFIDENTIALITÉ

Le coinjoin est une technique permettant de casser le traçage des bitcoins. Il repose sur une transaction collaborative à la structure spécifique de même nom : la transaction coinjoin. Les transactions coinjoin permettent d'améliorer la protection de la vie privée des utilisateurs de Bitcoin en rendant l'analyse des transactions plus difficile pour les observateurs extérieurs. Cette structure permet de mixer plusieurs pièces en une seule transaction, rendant difficile la détermination des liens entre les adresses d'entrée et de sortie.

Le fonctionnement général du coinjoin est le suivant : différents utilisateurs souhaitant mixer déposent un montant en input d'une transaction. Ces inputs ressortiront en différents outputs de même montant. À la sortie de la transaction, il est donc impossible de déterminer quel output appartient à quel utilisateur. Il n'y a techniquement aucun lien entre les entrées et les sorties de la transaction coinjoin. Le lien entre chaque utilisateur et chaque UTXO est cassé, de la même manière que l'historique de chaque pièce.



Pour permettre le coinjoin sans qu'aucun utilisateur ne perde la main sur ses fonds à aucun moment, la transaction est d'abord construite par un coordinateur puis transmise à chaque utilisateur. Chacun d'eux signe alors la transaction de son côté en vérifiant qu'elle lui convient, puis toutes les signatures sont ajoutées à la transaction. Si un utilisateur ou le coordinateur tente de voler les fonds des autres en modifiant les outputs de la transaction coinjoin, alors les signatures seront invalides et la transaction sera refusée par les nœuds. Lorsque l'enregistrement de l'output des participants se fait à l'aide de signatures aveuglées de Chaum pour éviter le lien avec l'input, on parle de « *Chaumian coinjoin* ».

Ce mécanisme augmente la confidentialité des transactions sans nécessiter de modifications du protocole Bitcoin. Des implémentations spécifiques de coinjoin, telles que Whirlpool, JoinMarket ou Wabisabi, proposent des solutions pour faciliter le processus de coordination entre les participants et renforcer l'efficacité de la transaction coinjoin. Voici une transaction coinjoin par exemple :

```
323df21f0b0756f98336437aa3d2fb87e02b59f1946b714a7b09df04d429dec2
```

Il est difficile de déterminer avec certitude qui a introduit en premier l'idée du coinjoin sur Bitcoin, et qui a eu l'idée d'utiliser les signatures aveugles de David

Chaum dans ce contexte. On pense souvent que c'est Gregory Maxwell qui en a parlé en premier dans un message sur BitcoinTalk en 2013 :

> *« En utilisant les signatures aveugles de Chaum : Les utilisateurs se connectent et fournissent des inputs (et des adresses de change) ainsi qu'une version cryptographiquement aveuglée de l'adresse vers laquelle ils souhaitent envoyer leurs pièces privées ; le serveur signe les jetons et les leur renvoie. Les utilisateurs se reconnectent anonymement, démasquent leurs adresses de sortie et les renvoient au serveur. Le serveur peut voir que toutes les sorties ont été signées par lui et que, par conséquent, toutes les sorties proviennent de participants valides. Plus tard, les personnes se reconnectent et signent. »*

Maxwell, G. (2013, 22 août). *CoinJoin : Bitcoin privacy for the real world*. BitcoinTalk Forum. https://bitcointalk.org/index.php?topic=279249.0

Toutefois, il existe d'autres mentions antérieures, à la fois pour les signatures de Chaum dans le cadre du mixage, mais également pour les coinjoins. En juin 2011, Duncan Townsend présente sur BitcoinTalk un mélangeur qui utilise les signatures de Chaum d'une manière assez similaire aux coinjoins chaumiens modernes. Dans le même thread, on peut retrouver un message de hashcoin en réponse à Duncan Townsend pour améliorer son mélangeur. Ce message présente justement ce qui ressemble le plus aux coinjoins. On retrouve également une mention d'un système similaire dans un message d'Alex Mizrahi en 2012, alors qu'il conseillait les créateurs de Tenebrix. Le terme en lui-même de « coinjoin » n'aurait pas été inventé par Greg Maxwell, mais il viendrait d'une idée de Peter Todd.

Le terme de « coinjoin » ne dispose pas de traduction française. Certains bitcoiners utilisent également les termes de « mix », de « mixing » ou encore de « mixage » pour évoquer la transaction coinjoin. Le mixage est plutôt le processus utilisé au cœur du coinjoin. Aussi, il ne faut pas confondre le mixage par coinjoins et le mixage par un acteur central qui prend possession des bitcoins durant le processus. Cela n'a rien à voir avec le coinjoin où l'utilisateur ne perd à aucun moment la main sur ses bitcoins durant le processus.

Terme associé : **CHAUMIAN COINJOIN**

COINJUMBLE

CONFIDENTIALITÉ

Logiciel développé par Chris Belcher et lancé en août 2014, conçu pour faciliter l'utilisation de coinjoins avec une *GUI*. À la différence d'autres implémentations de coinjoins de l'époque nécessitant que les participants souhaitent effectuer un coinjoin simultanément, CoinJumble permettait de partager les parties de transaction de manière asynchrone. Les utilisateurs pouvaient échanger via des canaux de communications externes pour partager les parties de transaction encodées. Aujourd'hui, CoinJumble n'est plus utilisé.

Termes associés : **CHAUMIAN COINJOIN** **COINJOIN**

COINKITE

ORGANISATION

Entreprise canadienne fondée en 2012, spécialisée dans la fabrication de matériel de sécurité pour Bitcoin. Coinkite est principalement connue pour le Coldcard, un *hardware wallet* conçu spécifiquement pour le stockage sécurisé de clés. Le Coldcard se distingue par son approche minimaliste centrée exclusivement sur Bitcoin, son fonctionnement air-gapped (déconnecté de tout réseau), et ses nombreuses fonctionnalités de sécurité avancées.

Coinkite développe également d'autres produits comme le Blockclock, une horloge qui affiche le prix du bitcoin et la hauteur de bloc, ainsi que le SatsCard et le TapSigner, des cartes NFC permettant de stocker ou de signer des transactions Bitcoin.

Termes associés : **COLD WALLET** | **HARDWARE WALLET**

COINMUX

CONFIDENTIALITÉ

Implémentation de coinjoin développée en 2014. Coinmux est un protocole de mixage de bitcoins décentralisé et trustless, qui ne nécessite ni la confiance entre les participants, ni l'intervention d'un acteur central. Le logiciel regroupe les bitcoins de plusieurs utilisateurs dans une transaction unique, où chaque sortie dispose de montants identiques, ce qui permet de casser les liens entre les inputs et les outputs. Le protocole de Coinmux assure que les utilisateurs conservent le contrôle de leurs fonds durant tout le processus, en ne faisant signer les transactions que lorsque les entrées et les sorties correspondent exactement à ce qui a été convenu. Aujourd'hui, ce protocole n'est plus utilisé.

Terme associé : **COINJOIN**

COINOS

OUTIL

Application web (PWA) qui permet d'envoyer, recevoir et accepter des paiements en bitcoins via on-chain, Lightning Network, Liquid et Ecash, directement depuis un navigateur. Coinos s'adresse aux particuliers et aux commerçants, avec des fonctionnalités de point de vente et une intégration Nostr. La plateforme est custodiale.

Termes associés : **BTCPAY SERVER** | **LIGHTNING NETWORK**

COINS/

PROTOCOLE

Nom de l'ancien dossier utilisé dans Bitcoin Core pour stocker l'UTXO set, remplacé par le dossier `chainstate/` dans la version 0.8.0.

Terme associé : **UTXO SET**

COINSHUFFLE

CONFIDENTIALITÉ

Protocole de mixage de bitcoins proposé en 2014 par Tim Ruffing, Pedro Moreno-Sanchez et Aniket Kate, inspiré de l'idée formalisée par Gregory Maxwell. CoinShuffle permet de couper l'historique de pièces sans nécessiter de tiers de confiance. Le protocole assure qu'aucun participant ne peut relier les entrées aux sorties des autres participants. Ce concept n'a jamais été largement adopté, les techniques de confidentialité telles que le *Chaumian coinjoin* lui étant préférées.

Terme associé : **COINJOIN**

COINSWAP

CONFIDENTIALITÉ

Protocole permettant un transfert secret de propriété entre des utilisateurs. Cette méthode vise à transférer la possession de bitcoins d'une personne à une autre, et inversement, sans que cet échange soit explicitement visible sur la blockchain. Le coinswap utilise des contrats intelligents pour faire le transfert sans besoin de confiance entre les parties.

Imaginons un exemple naïf (qui ne fonctionne pas) avec Alice et Bob. Alice détient 1 BTC sécurisé avec la clé privée A, et Bob en possède également 1, sécurisé avec la clé privée B. Ils pourraient théoriquement échanger leurs clés privées via un canal de communication externe pour réaliser un transfert secret. Cependant, cette méthode naïve présente un risque élevé en termes de confiance. Rien n'empêche Alice de conserver une copie de la clé privée A après l'échange et de l'utiliser ultérieurement pour dérober les bitcoins, une fois que la clé est entre les mains de Bob. De plus, il n'existe aucune garantie empêchant Alice de recevoir la clé privée B de Bob et de ne jamais lui transmettre sa clé privée A en échange. Cet échange repose donc sur une confiance excessive entre les parties et s'avère inefficace pour assurer un transfert secret de propriété de manière sécurisée.

Pour résoudre ces problèmes et permettre un coinswap entre parties qui ne se font pas confiance, on va plutôt utiliser des systèmes de contrats intelligents qui font en sorte que l'échange soit « atomique ». On peut utiliser des HTLC (*Hash Time-Locked Contracts*) ou des PTLC (*Point Time-Locked Contracts*). Ces deux protocoles fonctionnent de manière similaire en utilisant un système de verrouillage temporel qui garantit que l'échange est soit complété avec succès, soit entièrement annulé, ce qui permet de protéger l'intégrité des fonds des deux parties. La principale différence entre les HTLC et les PTLC réside dans le fait que les HTLC utilisent des hachages et des préimages pour sécuriser la transaction, tandis que les PTLC utilisent des Adaptor Signatures.

Dans un scénario de coinswap utilisant un HTLC ou un PTLC entre Alice et Bob, l'échange se déroule de manière sécurisée : soit il réussit et chacun reçoit le BTC de l'autre, soit il échoue et chacun conserve son propre BTC. Il est ainsi impossible pour l'une des parties de tricher ou de voler le BTC de l'autre.

L'utilisation des Adaptor Signatures est particulièrement intéressante dans ce

contexte, car elle permet de se passer des scripts traditionnels (c'est un mécanisme parfois désigné sous le terme de « *scriptless scripts* »). Cette caractéristique permet de réduire les frais associés à l'échange. Un autre avantage majeur des Adaptor Signatures est qu'elles ne nécessitent pas l'utilisation d'un hachage commun pour les deux parties de la transaction, ce qui évite ainsi de révéler un lien direct entre elles dans le cadre de certains types d'échanges.

Terme associé : **ADAPTOR SIGNATURE**

COLD WALLET

⊸ PORTEFEUILLE | FR : PORTEFEUILLE FROID

Synonyme de « hardware wallet ». Un hardware wallet, ou portefeuille matériel, est un dispositif électronique dédié à la sécurisation et à la gestion des clés privées d'un portefeuille Bitcoin. Ces périphériques sont conçus pour procurer une sécurité renforcée par rapport aux portefeuilles logiciels qui résident sur des machines polyvalentes et directement connectées à internet. Les hardware wallets stockent la phrase mnémonique hors ligne, sur un matériel qui dispose d'une infime surface d'attaque, ce qui l'isole des environnements potentiellement vulnérables.

Lorsqu'une transaction est effectuée, le portefeuille matériel signe la transaction à l'intérieur du dispositif lui-même, sans exposer la clé privée à l'extérieur. Une fois la transaction signée, elle est transmise au réseau Bitcoin pour être confirmée et incluse dans la blockchain. Parmi les modèles de hardware wallets les plus populaires, on peut citer : Ledger, Trezor, Coldcard, Passport, BitBox, Satochip, Jade ou encore SeedSigner (liste non exhaustive).

Le hardware wallet peut être exprimé de différentes manières en français. Certains parlent de « portefeuille matériel » ou bien de « portefeuille froid ». D'autres préfèrent que l'on emploie le terme de « périphérique de signature », ou « signing device » en anglais, afin d'éviter de faire penser que les bitcoins se trouvent physiquement dans le portefeuille.

Termes associés : **HARDWARE WALLET** | **PHRASE DE RÉCUPÉRATION**

COLDCARD

⊸ PORTEFEUILLE

Famille de *Hardware wallet* fabriqués par la société canadienne Coinkite, conçus exclusivement pour Bitcoin. Les Coldcard se distinguent par leur approche de sécurité maximale : ils sont entièrement *air-gapped*, c'est-à-dire qu'ils peuvent fonctionner sans jamais être connectés à un ordinateur, en échangeant les transactions via une carte microSD, NFC ou échange de QR codes. Les Coldcard intègrent des *secure elements* pour protéger les clés privées contre les accès physiques. Ils proposent de nombreuses fonctionnalités de sécurité avancées comme par exemple le « Brick Me PIN » (qui détruit les données) et le « Duress PIN » (qui ouvre un portefeuille leurre).

Il existe plusieurs Coldcard différentes :

- La sous-famille des MK (MK3, MK4), qui intègre un clavier numérique uniquement, un petit écran, et ne dispose pas d'appareil photo pour scanner les

QR codes ;
- La sous-famille des Q, le produit le plus haut de gamme de la marque, avec lecteur de QR codes, emplacement pour les piles et clavier QWERTY complet.

Termes associés : **HARDWARE WALLET** | **PSBT**

COLLATÉRAL

ÉCONOMIE ET RÉGULATION | EN : COLLATERAL

Actif donné en garantie par un emprunteur à un prêteur pour sécuriser un prêt ou une obligation financière. En cas de défaillance de l'emprunteur, le prêteur peut saisir le collatéral pour couvrir sa perte. Ce mécanisme réduit le risque de contrepartie et permet d'accéder à des conditions d'emprunt plus favorables.

Dans le contexte de Bitcoin, ce terme peut apparaître dans différentes situations. La plupart du temps, il est utilisé dans le cadre des prêts lombards. Des entités peuvent ainsi déposer leurs bitcoins en collatéral pour obtenir des prêts en monnaie *fiat*, ce qui leur permet de conserver leur exposition au bitcoin sans avoir à le vendre. Cela peut également permettre de vivre sur ses bitcoins sans les vendre, et donc sans déclencher d'événement fiscal sur les plus-values. Ce type de prêt lombard adossé au bitcoin est proposé par diverses plateformes.

Termes associés : **FIAT** | **LIGHTNING NETWORK**

COLORED COINS

COUCHE SUPÉRIEURE

Méthode proposée en 2012 par Yoni Assia, Vitalik Buterin et Meni Rosenfeld permettant de représenter et gérer des actifs non natifs sur la blockchain Bitcoin. L'idée était d'attacher des métadonnées à des transactions spécifiques, afin de colorer des bitcoins pour indiquer leur association avec des actifs physiques. La première implémentation fonctionnelle, le protocole Open Assets, a été développée par Flavien Charlon en 2013. Ce protocole permettait de marquer des bitcoins en utilisant des `OP_RETURN`.

Termes associés : **OAP - OPEN ASSETS PROTOCOL** | **OP_RETURN - 0X6A**

COMMERÇANT

PROTOCOLE | EN : MERCHANT

Toute personne physique ou morale qui accepte d'échanger un bien ou un service contre des bitcoins. Ce sont ces commerçants qui confèrent son utilité à la monnaie bitcoin. Plus une monnaie est acceptée par un large éventail de commerçants, plus elle devient utile pour les individus. Puisque les commerçants ont la capacité de déterminer l'utilité d'une monnaie en acceptant de l'échanger contre des biens et des services, dans le cas de Bitcoin, ils ont également un poids considérable dans le choix des règles de consensus. Chacun dispose d'un certain pouvoir proportionnel à l'activité économique qu'il est en capacité d'apporter à un fork. Parmi les commerçants, il y a évidemment les commerces, mais aussi les plateformes d'échange, les mineurs et les utilisateurs.

Termes associés : **FORK** | **RÈGLES DE CONSENSUS**

COMMIT

‹/› INFORMATIQUE

Dans le cadre de Git, représente une capture instantanée des modifications apportées à l'ensemble de fichiers d'un dépôt. Chaque commit est identifié par un hachage unique et inclut un message descriptif, l'identité de l'auteur et la date. Il permet de suivre l'évolution du projet et de revenir à des états antérieurs si nécessaire.

Termes associés : **CLONE** **GIT**

COMMITMENT

RGB | FR : ENGAGEMENT

Un Commitment (au sens cryptographique) est un objet mathématique, noté C, dérivé de façon déterministe à partir d'une opération sur une donnée structurée m (le message) et d'une valeur aléatoire r. On écrit :

$$C \;=\; commit(m,\; r)$$

Ce mécanisme comprend deux opérations principales :

- `Commit` : on applique une fonction cryptographique à un message m et à un aléa r pour produire C ;
- `Verify` : on utilise C, le message m et la valeur r pour vérifier que ce commitment est correct. La fonction renvoie `Vrai` ou `Faux`.

Un commitment doit respecter deux propriétés :

- `Binding` : il doit être impossible de trouver un couple (m', r') avec $m' \neq m$ tel que :

$$commit(m',\; r') \;=\; C$$

- `Hiding` : la connaissance de C ne doit pas révéler le contenu de m.

Dans le cas du protocole RGB, un commitment est inclus dans une transaction Bitcoin afin de prouver l'existence d'une certaine information à un instant donné, sans dévoiler cette information elle-même.

Termes associés : **CLIENT-SIDE VALIDATION** **STATE TRANSITION**

COMMITMENT FEE

LIGHTNING NETWORK

Frais inclus dans la transaction d'engagement (*commitment transaction*) d'un canal Lightning. Cette transaction permet à chaque participant de récupérer ses fonds en cas de fermeture unilatérale du canal. Pour être confirmée dans un délai raisonnable, elle doit comporter des frais de minage suffisants.

Avec l'introduction des *anchor channels*, le *commitment fee* peut être fixé à un montant très faible lors de la négociation du canal. En effet, les *anchor outputs* permettent d'augmenter les frais au moment de la diffusion effective de la transaction d'engagement en y attachant un UTXO supplémentaire via le mécanisme de CPFP (*Child Pays For Parent*). Cela évite de surestimer les frais

lors de la création du canal, tout en garantissant une confirmation rapide en cas de fermeture forcée.

Termes associés : **ANCHOR** | **TRANSACTION D'ENGAGEMENT**

COMPACT BLOCK FILTERS

RÉSEAU

Mécanisme permettant aux clients légers de vérifier si un bloc contient des transactions les concernant, sans télécharger l'intégralité de la blockchain et sans révéler leurs adresses à un nœud tiers. Contrairement aux filtres de Bloom du BIP-0037, où le client envoie un filtre à un nœud complet (exposant ainsi ses données), ce sont ici les nœuds complets qui génèrent des filtres déterministes pour chaque bloc. Le client les télécharge, les teste localement, puis récupère les blocs complets uniquement en cas de correspondance.

Les filtres sont construits selon le BIP-0158 à l'aide de *Golomb-Coded Sets* (GCS), une structure probabiliste plus compacte que les filtres de Bloom. Le protocole de distribution pair-à-pair est défini dans le BIP-0157, qui introduit des messages dédiés (`getcfilters`, `cfilter`, `getcfheaders`...) ainsi qu'un système d'en-têtes chaînés permettant de vérifier l'authenticité des filtres reçus. Un client disposant d'au moins un pair honnête peut ainsi détecter tout filtre invalide.

Ce mécanisme offre une meilleure confidentialité, puisque les blocs peuvent être téléchargés depuis n'importe quelle source, et réduit la charge de calcul imposée aux nœuds complets. En contrepartie, la bande passante consommée en fonctionnement normal est supérieure à celle du BIP-0037.

Termes associés : **BIP-0157** | **BIP-0158**

COMPACT BLOCK RELAY

RÉSEAU

Protocole introduit dans Bitcoin Core en 2016 via le BIP-0152 qui propose une méthode d'économie de bande passante pour les nœuds du réseau. *Compact Block Relay* permet de communiquer les informations des blocs de manière compacte, en se basant sur l'hypothèse que les nœuds ont déjà une grande partie des transactions d'un bloc récent dans leur mempool. Plutôt que de transmettre chaque transaction intégralement, ce qui constituerait un doublon, *Compact Block Relay* propose d'envoyer uniquement de courts identifiants pour les transactions déjà connues des pairs, accompagnés de quelques transactions sélectionnées (notamment la transaction coinbase et celles que le nœud est susceptible de ne pas connaître). Le nœud peut ensuite demander à ses pairs les éventuelles transactions manquantes. *Compact Block Relay* permet ainsi de diminuer la quantité de données échangées lors de la propagation des blocs, ce qui réduit ainsi les pics de bande passante et améliore l'efficacité globale du réseau.

Termes associés : **BIP-0152** | **MEMPOOL**

COMPACTSIZE

PROTOCOLE

Format d'encodage d'entiers non signés à longueur variable utilisé dans le protocole Bitcoin pour indiquer le nombre d'éléments ou la taille de certains champs dans les transactions et les messages réseau. Le CompactSize permet d'économiser de l'espace en utilisant moins d'octets pour les petites valeurs. L'encodage fonctionne comme suit :

- les valeurs de 0 à 252 sont encodées sur un seul octet ;

les valeurs de 253 à 65 535 sont précédées de l'octet `0×fd` *et encodées sur 2 octets en* little-endian* ;

- les valeurs de 65 536 à 4 294 967 295 sont précédées de `0×fe` et encodées sur 4 octets ;
- les valeurs supérieures sont précédées de `0×ff` et encodées sur 8 octets.

Le CompactSize est utilisé par exemple pour indiquer le nombre de transactions dans un bloc, le nombre d'entrées ou de sorties dans une transaction, ou encore la longueur d'un script.

Terme associé : **ENTÊTE DE BLOC**

COMPATIBILITÉ RÉTROSPECTIVE

PROTOCOLE | EN : BACKWARD COMPATIBILITY

Fait référence à la capacité d'une mise à jour des règles du protocole à maintenir la compatibilité avec les versions antérieures. Cela signifie que les modifications sont conçues de manière à ce que les anciens nœuds (les nœuds qui exécutent des versions antérieures au changement de règles) puissent toujours interagir avec le réseau et suivre la chaîne avec le plus de travail accumulé. Il faut donc que les anciens nœuds ne rejettent ni les nouveaux blocs, ni les nouvelles transactions.

La compatibilité rétrospective permet de réduire fortement la probabilité qu'une mise à jour fragmente le réseau, évitant ainsi la division des nœuds en sous-groupes sur des chaînes différentes. Pour assurer une compatibilité avec les versions antérieures du protocole, une mise à jour doit rendre les règles existantes plus strictes ou en introduire de nouvelles. C'est ce principe qui définit un « soft fork ». À l'inverse, si une mise à jour assouplit les règles existantes ou en élimine certaines, alors elle ne sera pas rétrocompatible. Ce sera donc un « hard fork ».

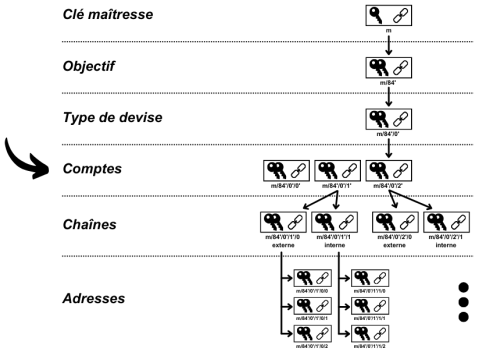
Termes associés : **HARD FORK** | **SOFT FORK**

COMPTE

PORTEFEUILLE | EN : ACCOUNT

Dans les portefeuilles HD (déterministes hiérarchiques), un compte représente une couche de dérivation à la profondeur 3 selon le BIP-0032. Chaque compte est numéroté séquentiellement à partir de `/0'/` (dérivation renforcée, donc en réalité `/2^31/` ou `/2 147 483 648/`). C'est à cette profondeur de dérivation que se trouvent les fameuses `xpub`. De nos jours, on utilise généralement un

seul compte au sein d'un portefeuille HD. Mais initialement, ils avaient été imaginés pour pouvoir ségréguer diverses catégories d'utilisation au sein d'un même portefeuille. Par exemple, si l'on prend un chemin de dérivation standard pour une adresse de réception Taproot externe (P2TR) : `m/86'/0'/0'/0/0`, l'index du compte est le second `/0'/`.



Terme associé : **CHEMIN DE DÉRIVATION**

CONCATÉNATION

‹/› INFORMATIQUE | EN : CONCATENATION

Dans le contexte de la cryptographie et des systèmes informatiques, désigne le processus d'assemblage de deux opérandes, en les mettant bout à bout, formant ainsi une nouvelle chaîne de caractères ou de données. Cette opération se note généralement avec un symbole de deux barres verticales $\|$, ou avec le symbole $\circ$. Par exemple, la concaténation de 45 avec 87 sera égale à 4587. Nous noterons : $45 \| 87 = 4587$. On a mis bout à bout les deux opérandes.

Termes associés : **FONCTION DE HACHAGE** | **OCTET**

CONDENSAT

CRYPTOGRAPHIE | EN : DIGEST / HASH

En cryptographie, désigne le résultat (ou l'*output*) produit par l'application d'une fonction de hachage cryptographique à un ensemble de données. Le condensat est une chaîne de caractères de taille fixe, généralement représentée sous forme d'une série de chiffres et de lettres en notation hexadécimale (base 16). Ce résultat a la particularité d'être presque unique et spécifique aux données d'entrée, de sorte qu'un changement minime dans l'entrée produira un condensat complètement différent. Les fonctions de hachage cryptographiques sont conçues pour être unidirectionnelles et résistantes aux collisions, rendant très difficile de retrouver les données initiales à partir du condensat ou de trouver deux entrées distinctes produisant le même condensat.

Terme associé : **FONCTION DE HACHAGE**

CONFIDENTIAL TRANSACTIONS

CRYPTOGRAPHIE | FR : TRANSACTIONS CONFIDENTIELLES

Protocole cryptographique implémenté sur le réseau Liquid qui permet de masquer les montants et les types d'actifs transférés dans une transaction, tout en permettant aux nœuds du réseau de vérifier sa validité. Les *Confidential Transactions* reposent sur des engagements de Pedersen (Pedersen *commitments*) pour dissimuler les valeurs, et sur des preuves de portée (*rangeproofs*) pour garantir que les montants engagés sont positifs et n'entraînent pas de création monétaire illégitime.

Ce mécanisme a été proposé par Adam Back en 2013, puis formalisé et implémenté par Gregory Maxwell. Les CT sont déployées sur Liquid depuis son lancement en 2018. Les premières versions utilisaient des signatures en anneau de Borromée (Borromean *ring signatures*) pour les preuves de portée, avant d'adopter des preuves plus efficaces comme les Bulletproofs.

Termes associés : **BORROMEAN RING SIGNATURES** | **ZKP - ZERO-KNOWLEDGE PROOF**

CONFIRMATION

PROTOCOLE

Correspond au nombre de blocs pour lesquels une transaction bénéficie de leur sécurité. Lorsque l'on diffuse une transaction au réseau Bitcoin, celle-ci est d'abord en attente dans les mempools des nœuds. Elle est ensuite incluse dans un bloc valide par un mineur. À ce stade, la transaction vient d'être ajoutée à la blockchain, elle bénéficie donc d'une première confirmation. Lorsqu'un nouveau bloc sera trouvé par-dessus le bloc où se trouve la transaction en question, elle bénéficiera d'une seconde confirmation, et ainsi de suite. Chaque nouveau bloc miné par-dessus le bloc contenant la transaction constitue une nouvelle confirmation. Grâce au comptage du nombre de confirmations pour une transaction, on peut estimer le risque qu'elle puisse être finalement annulée à cause d'une réorganisation. Le nombre de confirmations nous permet de juger du niveau d'immuabilité d'une transaction sur la blockchain.

Termes associés : **BLOC** | **MEMPOOL**

CONNECTOR

COUCHE SUPÉRIEURE | FR : CONNECTEUR

Sortie spéciale créée dans la transaction de round du protocole Ark pour garantir l'atomicité entre la cession des anciens VTXO (via les *forfeit transactions*) et la création des nouveaux. Sans ce mécanisme, un utilisateur signant une forfeit transaction remettrait le contrôle de son VTXO à l'ASP sans aucune assurance que le round aboutisse et que de nouveaux VTXO soient créés.

Concrètement, la transaction de round génère, en plus de l'arbre contenant les VTXO de sortie, un second arbre de covenants comportant une feuille par VTXO d'entrée. Ces feuilles sont les *connector outputs*. Elles ne portent pas de valeur économique significative, mais servent de lien cryptographique : en incluant un connector output comme entrée dans une forfeit transaction, celle-ci ne devient valide que si la transaction de round est confirmée on-chain. Si

le round échoue, la forfeit transaction est invalide et l'utilisateur conserve son VTXO d'origine. Ce mécanisme assure que les deux opérations (forfait et création) sont indissociables.

Termes associés : **FORFEIT TRANSACTION** | **ROUND ARK**

CONSENSUS

PROTOCOLE

Mécanisme par lequel tous les nœuds du réseau Bitcoin parviennent à s'accorder sur l'état partagé de la blockchain. Le consensus permet que tous les utilisateurs s'alignent sur un même historique des transactions Bitcoin, afin notamment d'éviter la double dépense. Le mécanisme de consensus de Bitcoin est parfois appelé « Consensus de Nakamoto ». Il s'appuie sur la preuve de travail et spécifie que tous les nœuds du réseau acceptent la chaîne disposant de la plus grande quantité de travail accumulé.

Par extension, certaines personnes appellent par « Consensus » les règles tacites du protocole Bitcoin.

Termes associés : **DOUBLE DÉPENSE** | **PREUVE DE TRAVAIL**

CONSENSUS CLEANUP

PROTOCOLE | FR : NETTOYAGE DU CONSENSUS

Proposition de soft fork (BIP-0054) regroupant plusieurs corrections de vulnérabilités anciennes dans les règles de consensus de Bitcoin. Plutôt que de déployer un soft fork par bug, le « Consensus Cleanup » les regroupe pour mutualiser le coût fixe d'un changement de consensus. La proposition, portée par Antoine Poinsot et Matt Corallo, cible quatre problèmes distincts.

Premièrement, l'attaque *time warp* : en manipulant les horodatages aux frontières des périodes d'ajustement de la difficulté, un attaquant majoritaire peut réduire artificiellement et injustement la difficulté à son minimum en quelques périodes. Le correctif impose que le premier bloc d'une période ait un horodatage au plus 2 heures avant celui du bloc précédent, et que le dernier bloc ait un horodatage au moins égal à celui du premier (pas de période négative en temps).

Deuxièmement, le temps de validation des blocs : des blocs conçus de manière spécifique peuvent prendre plus d'une heure à valider sur du matériel modeste. Une limite de 2 500 opérations de signature potentiellement exécutées par transaction réduit le pire cas d'un facteur 40.

Troisièmement, la faiblesse de l'arbre de Merkle : une transaction de 64 octets peut être confondue avec un nœud intermédiaire de l'arbre, permettant de falsifier des preuves d'inclusion. Ces transactions sont désormais invalides.

Quatrièmement, la duplication de transactions coinbase : certains blocs antérieurs au BIP-0034 contiennent un engagement de hauteur valide pour un bloc futur, ce qui pourrait créer des doublons à partir du bloc 1 983 702. Imposer un `nLockTime` égal à la hauteur moins un dans les coinbases élimine définitivement le besoin de validation BIP-0030. Ce BIP a le statut « Draft » pour le

moment.

Termes associés : **BIP-0054** **SOFT FORK**

CONSIGNMENT

RGB

Dans le cadre du protocole RGB, regroupe les données échangées entre les parties, soumises à la *Client-side Validation*. Il existe deux grandes catégories de *consignments* : `Contract Consignment` *: fourni par l'*issuer *(émetteur du contrat), il comprend les informations d'initialisation telles que le* Schema*, la* Genesis*, l'*Interface *et l'*Implementation *de l'*Interface*. `Transfer Consignment` *: fourni par la partie qui paie (*payer*). Il contient tout l'historique de transitions d'état aboutissant au* terminal consignment* (c'est-à-dire l'état final reçu par le bénéficiaire).

Ces *consignments* ne sont pas enregistrés publiquement dans la blockchain ; ils sont échangés directement entre les parties concernées sur le canal de communication de leur choix.

Termes associés : **CLIENT-SIDE VALIDATION** **GENESIS**

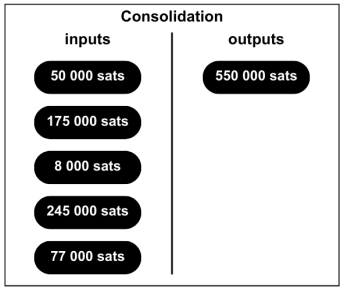
CONSOLIDATION

PORTEFEUILLE

Transaction spécifique dans laquelle plusieurs petits UTXOs sont fusionnés en entrée pour former un seul et plus gros UTXO en sortie. Cette opération est une transaction effectuée vers son propre portefeuille. L'objectif de la consolidation est de tirer profit des périodes où les frais sur le réseau Bitcoin sont bas pour fusionner plusieurs petits UTXOs en un seul plus grand en valeur. Ainsi, on anticipe les dépenses obligatoires en cas de hausse des frais, permettant d'économiser sur les frais de transaction futurs.

En effet, les transactions comportant de nombreuses entrées sont plus lourdes et, par conséquent, plus coûteuses. Outre l'économie réalisable sur les frais de transaction, la consolidation est aussi une forme de planification à long terme. Si votre portefeuille contient de très petits UTXOs, ceux-ci peuvent devenir inutilisables si le réseau Bitcoin entre dans une période prolongée de frais élevés. Par exemple, si vous devez dépenser un UTXO de 10 000 sats mais que les frais de minage minimums s'élèvent à 15 000 sats, la dépense excéderait la valeur de l'UTXO lui-même. Ces petits UTXOs deviennent alors économiquement non rationnels à utiliser et restent inutilisables tant que les frais ne baissent pas. Ces UTXOs sont communément appelés « *dust* » (poussière). En consolidant régulièrement vos petits UTXOs, vous réduisez ce risque associé aux augmentations de frais.

Cependant, il est important de noter que les transactions de consolidation sont reconnaissables lors d'une analyse de chaîne. Une telle transaction indique une CIOH (*Common Input Ownership Heuristic*), c'est-à-dire que les entrées de la transaction de consolidation sont possédées par une seule entité. Cela peut avoir des implications en termes de confidentialité pour l'utilisateur.



Termes associés : **CIOH** **UTXO**

CONTENEUR

↗ MINAGE | EN : CONTAINER

Dans le cadre du minage, un conteneur est une structure modulaire utilisée pour héberger et opérer un grand nombre d'ASICs. Ces conteneurs sont conçus pour optimiser l'espace, la gestion de la chaleur et l'alimentation électrique. Ils peuvent être équipés de différents systèmes de refroidissement, tels que l'*air cooling*, l'*hydro cooling* ou l'*immersion cooling*. Leur avantage principal réside dans leur mobilité et leur capacité à être déployés rapidement, souvent à proximité de sources d'énergie bon marché.

Termes associés : **AIR COOLING** **ASIC**

CONTRACT OPERATION

RGB

Dans le cadre du protocole RGB, désigne une mise à jour de l'état du contrat effectuée selon les règles du Schema. Les opérations suivantes existent dans RGB :

- State Transition ;
- Genesis ;
- State Extension.

Chaque opération modifie l'état en y ajoutant ou en y remplaçant certaines données (Global State, Owned State…).

Termes associés : **GENESIS** **STATE TRANSITION**

CONTRACT PARTICIPANT

RGB

Dans le cadre de RGB, un *Contract Participant* est un acteur prenant part aux opérations relatives au contrat. On distingue ainsi : *L'*issuer *du contrat, qui crée la* Genesis* (l'origine du contrat) ; *Les* contract parties*, c'est-à-dire les détenteurs de droits sur l'état du contrat ; *Les* public parties*, acteurs pouvant construire des* State Extensions *si le contrat propose des* Valencies* accessibles au public.

Termes associés : **CONTRAT RGB** **STATE EXTENSION**

CONTRACT RIGHTS

RGB

Les *Contract Rights* désignent les différents droits que peuvent exercer les acteurs d'un contrat RGB. Ils se classent en plusieurs catégories : *Les* ownership rights*, associés à la détention d'un UTXO particulier (via un Seal Definition) ; *Les* executive rights*, c'est-à-dire la capacité de construire une ou plusieurs transitions (State Transitions) conformes au Schema ; *Les* public rights*, lorsque le Schema autorise certains usages publics, par exemple la création d'une State Extension via la rédemption d'une Valency.

Termes associés : **SEAL DEFINITION** | **STATE TRANSITION**

CONTRACT STATE

RGB

Dans le cadre de RGB, le *Contract State* correspond à l'état courant d'un contrat à un instant donné. Il peut être constitué de données à la fois publiques et privées, qui reflètent la situation du contrat. Dans RGB, on distingue : *Le* Global State*, qui comprend les propriétés publiques du contrat (mises en place dès la* Genesis* ou ajoutées via des mises à jour autorisées) ; *Les* Owned States*, qui appartiennent à des détenteurs précis, identifiés par leurs UTXOs.

Termes associés : **GLOBAL STATE** | **OWNED STATE**

CONTRAT INTELLIGENT

SCRIPT | EN : SMART CONTRACT

Programme qui s'exécute automatiquement lorsque certaines conditions prédéfinies sont remplies. Un contrat intelligent est donc un ensemble de clauses entre plusieurs parties qui peuvent se réaliser sans nécessiter l'intervention d'un tiers de confiance. Ces contrats déclenchent généralement des actions spécifiques comme un transfert de bitcoins.

En français, on parle également parfois de « contrat autonome ».

Termes associés : **CONTRAT RGB** | **SCRIPT**

CONTRAT RGB

RGB

Désigne un ensemble de droits exécutés numériquement entre plusieurs acteurs via le protocole RGB. Il possède un état actif et une logique d'affaires, définie par un *Schema*, qui précise quelles opérations sont autorisées (transferts, extensions, etc.). L'état d'un contrat, ainsi que les règles de validité, s'expriment dans le *Schema*. À tout moment, le contrat n'évolue que conformément à ce qui est permis par ce *Schema* et par les scripts de validation (exécutés, par exemple, dans AluVM).

Termes associés : **ALUVM** | **SCHEMA**

CONTRIBUTEUR - CORE

COMMUNAUTÉ | EN : CONTRIBUTOR

Un contributeur dans le contexte de Bitcoin Core (l'implémentation majoritaire de nœuds sur le réseau Bitcoin) est une personne qui participe activement au développement du logiciel en écrivant du code, en examinant et en testant les modifications proposées par d'autres. Contrairement aux mainteneurs, les contributeurs n'ont pas le pouvoir de fusionner les modifications dans le code principal ; leur rôle est plutôt de soumettre des *pull requests* (PR) et de participer à la discussion et à la validation de ces propositions. Tout individu intéressé peut devenir contributeur sans nécessité d'une nomination ou d'une approbation formelle, ce qui diffère des mainteneurs qui sont chargés de responsabilités administratives et de décision plus élevées dans le projet.

Termes associés : **BITCOIN CORE** | **PULL REQUEST**

CONTROL BLOCK

SCRIPT | FR : BLOC DE CONTROLE

Structure de données fournie dans le témoin d'une transaction P2TR lors d'une dépense par chemin de script (*script path spend*), telle que définie dans le BIP-0341. Le control block prouve qu'un script donné fait bien partie de l'arbre de scripts (MAST) qui a servi à construire la clé publique Taproot de la sortie.

Il se compose de trois éléments : *un premier octet qui encode la version du script feuille (*leaf version*) et la parité de la clé publique de sortie ; *la clé publique interne (*internal key*) sur 32 octets, qui est la clé avant application du tweak ;

- le chemin de Merkle, constitué de zéro à 128 hachages de 32 octets chacun, qui représente la suite des nœuds frères nécessaires pour recalculer la racine de l'arbre depuis la feuille révélée.

Lors de la validation, le nœud recalcule la racine de Merkle à partir du script feuille et du chemin fourni, puis vérifie que le tweak de la clé publique interne avec cette racine correspond bien à la clé publique de sortie inscrite dans le `scriptPubKey`. Ce mécanisme permet de prouver l'appartenance d'un script à l'arbre sans révéler les autres feuilles, ce qui préserve la confidentialité des chemins de dépense non utilisés.

Termes associés : **BIP-0341** | **P2TR** | **TAPROOT**

CONTROL BOARD

MINAGE | FR : CARTE DE CONTRÔLE

Composant dans une machine de minage qui permet de gérer et de coordonner les opérations de l'appareil. La carte de contrôle gère la communication entre le logiciel et les puces ASICs. Elle permet également de relever les performances du matériel et de contrôler sa température.

Termes associés : **ASIC** | **HASHRATE**

COOKIE

PROTOCOLE

Fichier utilisé pour l'authentification RPC (*Remote Procedure Call*) dans Bitcoin Core. Lorsque bitcoind démarre, il génère un cookie d'authentification unique et le stocke dans ce fichier. Les clients ou les scripts qui souhaitent interagir avec bitcoind via l'interface RPC peuvent utiliser ce cookie pour s'authentifier de manière sécurisée. Ce mécanisme permet une communication sûre entre le bitcoind et les applications externes (comme les logiciels de portefeuille par exemple), sans nécessiter une gestion manuelle des noms d'utilisateur et des mots de passe. Le fichier `.cookie` est régénéré à chaque redémarrage de bitcoind et supprimé à l'arrêt.

Termes associés : **BITCOIN CORE** | **BITCOIND**

COOPERATIVE CLOSE

LIGHTNING NETWORK | FR : FERMETURE COOPÉRATIVE

Fermeture d'un canal Lightning réalisée avec l'accord mutuel des deux parties. C'est la méthode de fermeture la plus avantageuse : les deux partenaires du canal signent conjointement une transaction de clôture qui redistribue les fonds vers leurs adresses respectives, sans timelock.

Contrairement à une fermeture forcée (*force close*), où une transaction d'engagement unilatérale est diffusée et impose un *timelock* avant que les fonds ne soient accessibles, la fermeture coopérative produit une transaction simple. Les frais de minage sont donc réduits (puisqu'ils peuvent être ajustés au marché de frais du moment), et les fonds sont disponibles dès la confirmation de la transaction.

Termes associés : **CANAL DE PAIMENT** | **FORCE CLOSE**

COORDINATEUR DE COINJOIN

CONFIDENTIALITÉ | EN : COINJOIN COORDINATOR

Serveur central qui met en relation les utilisateurs souhaitant faire des coinjoins et qui coordonne la construction des transactions collaboratives, sans pour autant pouvoir associer les inputs aux outputs spécifiques de chaque utilisateur. Cela est possible grâce à l'utilisation des signatures aveugles de Chaum et du réseau Tor. Contrairement à un mélangeur de bitcoins traditionnel, le coordinateur de coinjoin ne prend jamais possession des bitcoins des participants et n'est pas capable d'associer les inputs et les outputs.

Terme associé : **CHAUMIAN COINJOIN**

COUNTERPARTY

COUCHE SUPÉRIEURE

Méta-protocole open source lancé en janvier 2014 qui permettait de créer et d'échanger des actifs numériques directement sur la blockchain Bitcoin. Counterparty encode les métadonnées des tokens dans les transactions Bitcoin, principalement via des sorties multisig ou `OP_RETURN`, sans modifier le protocole de base.

Termes associés : **OP_RETURN - 0X6A** | **ORDINALS THEORY**

COURBE ELLIPTIQUE

CRYPTOGRAPHIE | EN : ELLIPTIC CURVE

Dans le contexte de la cryptographie, une courbe elliptique est une courbe algébrique définie par une équation de la forme $y^2 = x^3 + ax + b$. Ces courbes sont utilisées dans la cryptographie à courbes elliptiques (ECC), qui est une méthode de cryptographie à clé publique permettant de créer des algorithmes de chiffrement, de signature numérique et d'échange de clés. Dans le contexte de Bitcoin, l'algorithme ECDSA (*Elliptic Curve Digital Signature Algorithm*) ou le protocole de Schnorr sont utilisés avec la courbe `secp256k1`. Cette courbe a été choisie pour ses propriétés de performance et de sécurité. Ces algorithmes sont utilisés pour générer des clés publiques à partir de clés privées, ainsi que pour signer des transactions, et donc débloquer des bitcoins.

Termes associés : **ECDSA** | **SECP256K1**

COURS LÉGAL

ÉCONOMIE ET RÉGULATION | EN : LEGAL TENDER

Statut juridique attribué à une monnaie par un État, qui oblige tout créancier à l'accepter en paiement d'une dette. Une monnaie à cours légal ne peut pas être refusée pour le règlement d'une obligation libellée dans la monnaie nationale.

En septembre 2021, le Salvador est devenu le premier pays à conférer le cours légal au bitcoin aux côtés du dollar américain. Cette décision a suscité des réactions contrastées : certains bitcoiners y voient un précédent pour l'adoption de Bitcoin, tandis que d'autres ont critiqué l'obligation d'acceptation imposée aux commerçants, depuis abandonnée.

Termes associés : **FIAT** | **LOI DE GRESHAM**

COVENANT

PROTOCOLE

Mécanisme qui permet d'imposer des conditions spécifiques sur la manière dont une pièce donnée peut être dépensée, y compris dans des transactions futures. Au-delà des conditions usuellement autorisées par le langage script sur un UTXO, le covenant force des contraintes supplémentaires sur la manière de dépenser cette pièce Bitcoin dans des transactions ultérieures. Techniquement, l'instauration d'un covenant intervient lorsque le `scriptPubKey` d'un UTXO définit des restrictions sur le `scriptPubKey` des sorties d'une transaction qui dépense ledit UTXO. En élargissant la portée de script, les covenants permettraient de nombreuses évolutions sur Bitcoin comme l'ancrage bilatéral des drivechains, la mise en place de vaults ou encore l'amélioration des systèmes de surcouche comme Lightning. On différencie les propositions de covenants en fonction de trois critères :

- Leur portée ;
- Leur implémentation ;
- Leur récursivité.

Il existe de très nombreuses propositions qui permettraient l'utilisation de covenants sur Bitcoin. Les plus avancées dans le processus d'implémentation sont : `OP_CHECKTEMPLATEVERIFY` (CTV), `SIGHASH_ANYPREVOUT` (APO) et `OP_VAULT`. Parmi les autres propositions, il y a : `OP_TX`, `OP_TAPLEAFUPDATEVERIFY` (TLUV), `OP_EVICT`, `OP_CHECKSIGFROMSTACK` (CSFS), `OP_CAT`, etc.

Pour bien comprendre le concept de covenant, je vous propose une analogie : imaginez un coffre-fort contenant 500 € en petites coupures. Si vous parvenez à déverrouiller ce coffre avec la clé adéquate, alors vous êtes libre d'utiliser cet argent comme bon vous semble. Ça, c'est la situation normale de Bitcoin. Maintenant, imaginez que ce même coffre-fort ne contient pas 500 € en billets de banque, mais plutôt des tickets restaurants d'une valeur équivalente. Si vous réussissez à ouvrir ce coffre, vous pouvez disposer de cette somme. Cependant, votre liberté de dépense est restreinte, car vous ne pouvez utiliser ces tickets pour payer que dans certains restaurants. Ainsi, il y a une première condition pour dépenser cet argent, qui est de parvenir à ouvrir le coffre avec la clé appropriée. Mais il y a aussi une condition supplémentaire quant à l'usage futur de cette somme : elle doit être dépensée exclusivement dans des restaurants partenaires, et non pas en toute liberté. Ce système de contraintes sur les transactions futures, c'est ce que l'on appelle un covenant.

En français, il n'existe aucun terme pour capturer précisément la signification du mot « covenant ». On pourrait parler de « clause », de « pacte » ou d' « engagement », mais cela ne correspondrait pas exactement au terme « covenant ». Ce dernier est d'ailleurs emprunté d'une terminologie juridique qui permet de décrire une clause contractuelle imposant des obligations persistantes sur un bien.

Termes associés : DRIVECHAIN | SCRIPTPUBKEY

COVERT ASICBOOST

↗ MINAGE

Version secrète d'AsicBoost. AsicBoost est une méthode d'optimisation algorithmique pour le minage de Bitcoin. Dans sa version Covert, les mineurs manipulent l'arbre de Merkle plutôt que le Nonce, ce qui réduit ainsi les calculs nécessaires pour chaque hachage SHA256 en conservant certaines données inchangées entre les tentatives de hachage. Contrairement à la version Overt d'AsicBoost, la version Covert dissimule l'utilisation d'AsicBoost durant le processus de minage. Cependant, depuis l'introduction de SegWit et son second arbre de Merkle, cette méthode n'est plus efficace, car le nombre de calculs requis pour son utilisation est devenu trop important par rapport à un processus de minage classique.

Termes associés : **ASICBOOST** | **OVERT ASICBOOST**

CPFP - CHILD PAY FOR PARENT

⊸ PORTEFEUILLE

Mécanisme transactionnel visant à accélérer la confirmation d'une transaction Bitcoin, tout comme le fait *Replace-by-Fee* (RBF), mais du côté du destinataire. Lorsqu'une transaction avec des frais trop faibles par rapport au marché reste bloquée dans les mempools des nœuds et ne se confirme pas assez rapidement, le destinataire peut faire une nouvelle transaction, en dépensant les bitcoins reçus dans la transaction bloquée, bien qu'elle ne soit pas encore confirmée. Cette seconde transaction nécessite forcément que la première soit minée pour être confirmée. Les mineurs sont donc obligés d'inclure les deux transactions ensemble. La seconde va allouer beaucoup plus de frais de transaction que la première, de telle sorte que la moyenne de frais incite les mineurs à inclure les deux transactions. La transaction enfant (la seconde) paie pour la transaction parent qui est bloquée (la première). C'est pour cela que l'on parle d'un « CPFP ».

Ainsi, CPFP permet au destinataire d'obtenir plus rapidement ses fonds malgré les faibles frais initiaux engagés par l'expéditeur, contrairement à RBF (*Replace-by-Fee*) qui permet à l'envoyeur de prendre l'initiative d'accélérer sa propre transaction en augmentant les frais.

Termes associés : **RBF - REPLACE-BY-FEE** | **TRANSACTION NON CONFIRMÉE**

CPPSRB

↗ MINAGE

Sigle de « *Capped Pay Per Share with Recent Backpay* ». C'est une méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. Dans ce système, la pool paie autant de shares qu'elle le peut à chaque fois qu'un bloc est trouvé, en donnant la priorité aux shares les plus récentes. Cette méthode permet de garantir la stabilité financière de la pool de minage, tout en offrant une rémunération à la tâche et en incitant les mineurs à rester connectés à la pool pour éviter le *pool hopping*.

Terme associé : **SHARES**

CPU - CENTRAL PROCESSING UNIT

</> INFORMATIQUE | FR : PROCESSEUR

Composant principal d'un ordinateur responsable de l'exécution des instructions machines des logiciels. Dans le contexte de Bitcoin, le CPU était initialement utilisé pour le minage par les nœuds avant d'être surpassé par le minage par GPU (cartes graphiques), puis par l'utilisation de puces spécialisées que l'on appelle des « ASIC ».

Termes associés : **ASIC** **MINAGE**

CRYPTANALYSE

CRYPTOGRAPHIE | EN : CRYPTANALYSIS

Étude des techniques mathématiques pour tenter de casser les techniques cryptographiques. Cela inclut les processus de recherche d'erreurs ou de faiblesses dans l'implémentation d'une méthode cryptographique ou dans la méthode cryptographique elle-même.

Termes associés : **CRYPTOGRAPHIE** **CRYPTOLOGIE**

CRYPTER

CRYPTOGRAPHIE

Ce terme n'existe pas. On dit « chiffrer ».

Terme associé : **CRYPTOGRAPHIE**

CRYPTO-ACTIF

$ ÉCONOMIE ET RÉGULATION | EN : CRYPTO-ASSET

Terme utilisé dans un contexte juridique et réglementaire pour désigner les divers types de cryptomonnaies, dont le bitcoin.

Termes associés : **ALTCOIN** **CRYPTOMONNAIE**

CRYPTOGRAPHIE

CRYPTOGRAPHIE | EN : CRYPTOGRAPHY

Discipline qui incarne les principes, les moyens et les méthodes de transformation des informations, notamment avec des techniques mathématiques, afin de masquer leur contenu sémantique, d'empêcher leur utilisation non autorisée, d'assurer leur authenticité ou d'empêcher leur modification non détectée. La cryptographie regroupe l'utilisation de fonctions de hachage, de signatures numériques et d'algorithmes de chiffrement.

Termes associés : **FONCTION DE HACHAGE** **SIGNATURE NUMÉRIQUE**

CRYPTOLOGIE

CRYPTOGRAPHIE | EN : CRYPTOLOGY

Science mathématique qui traite de la cryptanalyse et de la cryptographie.

Termes associés : **CRYPTANALYSE** **CRYPTOGRAPHIE**

CRYPTOMONNAIE

ÉCONOMIE ET RÉGULATION | EN : CRYPTOCURRENCY

Qualificatif générique donné à toute forme de monnaie, d'actif, de crédit ou d'unité numérique au sein d'un système informatique dans lequel on utilise de la cryptographie pour les échanges et les transactions entre les utilisateurs.

Termes associés : **CRYPTO-ACTIF** | **CRYPTOGRAPHIE**

CTUSD - CITREA USD

COUCHE SUPÉRIEURE

Stablecoin natif de la plateforme Citrea, adossé au dollar américain à parité 1 :1. Le ctUSD (*Citrea USD*) est garanti par des bons du Trésor américain à court terme et des réserves en espèces. Il est émis en partenariat avec MoonPay et repose sur l'infrastructure de M0, un protocole de stablecoin institutionnel.

Termes associés : **CITREA** | **STABLECOIN**

CURTAILMENT

MINAGE | FR : RÉDUCTION - MINAGE

Désigne la réduction volontaire de la consommation d'électricité d'une ferme de minage en éteignant des machines. Cette réduction fait partie de certains contrats passés avec des producteurs d'électricité. Lorsqu'il y a une surproduction d'énergie par rapport à la demande du marché, les mineurs utilisent ce surplus. Lorsqu'il y a des pics de demande par rapport à la production, les mineurs peuvent réduire leur consommation afin de stabiliser le réseau.

Termes associés : **FERME DE MINAGE** | **PPA**

CUSTODY

ÉCONOMIE ET RÉGULATION

Dans le contexte de Bitcoin, se réfère à la détention et à la gestion des clés privées qui permettent le contrôle de bitcoins. La *custody* de BTC peut être assurée de deux manières : soit personnellement par l'utilisateur, qui garde lui-même les clés privées nécessaires pour accéder à ses bitcoins (ce que l'on appelle « *self-custody* »), soit par un tiers, comme une plateforme d'échange, où l'entreprise détient les clés privées et gère les bitcoins au nom de l'utilisateur. Cette seconde option est plus risquée que la *self-custody*, car cela expose les fonds de l'utilisateur aux risques de piratage, de faillite ou de comportements frauduleux de la part du gestionnaire.

En français, on utilise généralement le terme anglais de « custody ». On pourrait également le traduire par « garde ».

Termes associés : **CLÉ PRIVÉE** | **SELF-CUSTODY**

CYPHERPUNKS

HISTOIRE

Communauté informelle et internationale de personnes promouvant l'utilisation de la cryptographie comme moyen d'assurer les libertés individuelles sur Internet. Les cypherpunks défendent le droit fondamental pour l'individu de protéger sa vie privée, en particulier dans un contexte d'augmentation de la surveillance étatique et d'exploitation des données par des entités privées. Leur nom est issu des mots anglais *cypher*, « chiffre » (dans le sens de code secret), et *punk*. Il s'agit d'un calque du terme *cyberpunk*, qui désigne un genre littéraire décrivant un futur dystopique où la technologie de pointe se marie au vice et au crime d'une société déliquescente.

Le mouvement des cypherpunks remonte au début des années 1990, lorsque des groupes de cryptographes, de programmeurs et de libertariens ont commencé à discuter des implications politiques de la cryptographie au cours de leurs premières réunions dans la Silicon Valley. La personnalité principale au sein de la communauté était Tim May, qui avait écrit le *Crypto Anarchist Manifesto* en 1988, texte fondateur du mouvement. Parmi les moments clés de l'histoire des cypherpunks, il y a la fondation, en 1992, de la « Cypherpunks mailing list », une liste de diffusion de courrier électronique qui a servi pour ces discussions. La publication par Eric Hughes en 1993 du *A Cypherpunk's Manifesto*, document exposant les objectifs et les actions des cypherpunks, a également été un moment important.

L'idée d'une monnaie électronique qui ne s'établit pas sur une entité centrale, comme Bitcoin, est enracinée dans la philosophie des cypherpunks. La création de Bitcoin est souvent considérée comme une réalisation majeure de cette vision, aux côtés de PGP, Tor et WikiLeaks.

Termes associés : B-MONEY | TOR

D

DAEMON

</> INFORMATIQUE

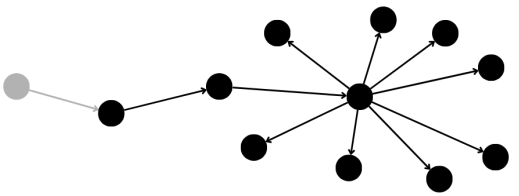
Type de programme informatique qui fonctionne en arrière-plan, indépendamment du contrôle de l'utilisateur. Ces programmes effectuent des tâches telles que la gestion de services réseau, la surveillance de systèmes, et la réponse à des requêtes sans nécessiter d'intervention directe. Dans le contexte de Bitcoin Core, le daemon s'appelle « `bitcoind` ». Il représente la version de Bitcoin Core sans interface utilisateur, qui fonctionne exclusivement en ligne de commande.

Termes associés : **BITCOIN CORE** **BITCOIND**

DANDELION

CONFIDENTIALITÉ

Proposition qui vise à améliorer la confidentialité du routage des transactions dans le réseau Bitcoin pour contrer la désanonymisation. Dans le fonctionnement classique de Bitcoin, les transactions sont immédiatement diffusées à de multiples nœuds. Ce phénomène peut potentiellement permettre à des observateurs de lier des transactions, normalement anonymes, avec des adresses IP. L'objectif du BIP-0156 est de traiter ce problème. Pour ce faire, il introduit une phase supplémentaire dans la diffusion permettant de préserver l'anonymat avant la propagation publique. Ainsi, Dandelion utilise d'abord une phase de « tige » où la transaction est envoyée à travers un chemin aléatoire de nœuds, avant d'être diffusée à l'ensemble du réseau dans la phase de « capitule ». La tige et le capitule sont des références au comportement de la propagation de la transaction à travers le réseau, qui ressemble à la forme d'un pissenlit (« *a dandelion* » en anglais). Cette méthode de routage brouille la piste menant au nœud source, rendant difficile de retracer une transaction via le réseau jusqu'à son origine.



Terme associé : **BIP-0156**

DARKWALLET

CONFIDENTIALITÉ

Logiciel de portefeuille Bitcoin axé sur la confidentialité, lancé par Amir Taaki et Cody Wilson en 2014, fonctionnant comme une extension pour le navigateur Google Chrome. DarkWallet disposait de fonctionnalités pour améliorer la confidentialité de l'utilisateur de Bitcoin, telles que les paiements furtifs et les *coinjoins*. Son développement a été abandonné depuis janvier 2015.

Termes associés : **COINJOIN** **SILENT PAYMENTS**

DARWINISME MONÉTAIRE

ÉCONOMIE ET RÉGULATION | EN : MONETARY DARWINISM

Concept qui applique la théorie de la sélection naturelle de Darwin au domaine monétaire. Il postule que les différentes formes de monnaie sont en compétition les unes avec les autres et que les monnaies possédant les propriétés les plus adaptées (rareté, durabilité, divisibilité, portabilité, fongibilité, vérifiabilité) finissent par supplanter celles qui sont moins performantes.

Dans cette perspective, les partisans de ce concept perçoivent Bitcoin comme une monnaie supérieure susceptible de s'imposer face aux monnaies fiat par la seule force de ses propriétés intrinsèques. Son offre strictement limitée, sa résistance à la censure et sa nature décentralisée constitueraient des avantages compétitifs décisifs dans cette sélection monétaire.

Ce concept est étroitement lié à celui d'hyper-bitcoinisation et s'inscrit dans la lignée de la loi de Thiers (la bonne monnaie chasse la mauvaise), en opposition à la loi de Gresham (la mauvaise monnaie chasse la bonne).

Termes associés : **FIAT** | **HYPER-BITCOINISATION**

DATABASE/

PROTOCOLE

Ancien dossier contenant des bases de données pour le portefeuille Bitcoin Core. Depuis la version 0.16, cette base de données a été déplacée dans le dossier `wallets/`.

Termes associés : **BERKELEYDB** | **WALLET.DAT**

DB.LOG

PROTOCOLE

Ancien fichier d'erreur BDB (*BerkeleyDB*) du portefeuille Bitcoin Core déplacé dans le dossier `wallets/` depuis la version 0.16.

Termes associés : **DATABASE/** | **WALLET.DAT**

DBC

RGB

Sigle de « Deterministic Bitcoin Commitments ». C'est un ensemble de règles permettant d'inscrire de manière prouvable et unique un engagement dans une transaction Bitcoin. Dans le protocole RGB, il existe deux formes principales de DBC :

- `Opret` ;
- `Tapret`.

Ces mécanismes définissent précisément comment l'engagement est encodé dans les sorties ou dans la structure d'une transaction Bitcoin, afin de s'assurer que cet engagement est repérable et vérifiable de façon déterministe.

Termes associés : **COMMITMENT** | **RGB**

DCA - DOLLAR COST AVERAGING

ÉCONOMIE ET RÉGULATION

Stratégie d'investissement qui consiste à investir un montant fixe de monnaie fiat dans un actif spécifique à intervalles réguliers, indépendamment du prix de l'actif. Cette méthode permet de réduire l'impact de la volatilité sur l'investissement global en achetant plus d'unités quand les prix sont bas et moins quand ils sont élevés. Le DCA vise généralement à construire une position sur un actif sur le long terme en minimisant les risques. C'est une stratégie très populaire pour l'investissement dans le bitcoin.

Termes associés : **BEAR MARKET** | **HODL**

DDOS

ATTAQUE | FR : ATTAQUE PAR DÉNI DE SERVICE DISTRIBUÉ

Forme de DOS où l'attaque provient de multiples sources simultanément, rendant la défense plus complexe. Les attaquants utilisent souvent des réseaux d'ordinateurs infectés par des malwares (botnets) pour lancer des requêtes massives vers une seule cible. Cette stratégie multiplie l'efficacité de l'attaque en surchargeant les capacités du système ciblé afin de provoquer des interruptions de service.

Termes associés : **DOS - DENIAL OF SERVICE** | **SYBIL**

DEBUG.LOG

PROTOCOLE

Fichier contenant l'historique des événements de Bitcoin Core. Il contient des données de journalisation, telles que les messages d'erreur, les avertissements et d'autres informations de débogage. Ce fichier est utilisé pour résoudre d'éventuels problèmes techniques.

Termes associés : **BITCOIN CORE** | **BITCOIND**

DÉCENTRALISATION

RÉSEAU | EN : DECENTRALIZATION

Principe d'organisation d'un système dans lequel aucune entité unique ne détient le contrôle ni l'autorité sur l'ensemble du fonctionnement. Un système décentralisé repose sur un ensemble de participants indépendants qui collaborent selon des règles communes, sans point de défaillance unique ni besoin de confiance envers un tiers central.

Dans Bitcoin, la décentralisation se manifeste à plusieurs niveaux. Le réseau est composé de milliers de nœuds répartis à travers le monde, chacun vérifiant indépendamment les transactions et les blocs selon les règles du protocole. Le minage est assuré par de nombreux acteurs en compétition, et aucun mineur ne peut à lui seul imposer des modifications. Le développement du logiciel est ouvert et collaboratif, sans autorité décisionnaire unique.

Termes associés : **CONSENSUS** | **MINAGE** | **NOEUD**

DÉFLATION

Ⓢ ÉCONOMIE ET RÉGULATION | EN : DEFLATION

Phénomène économique caractérisé par une baisse générale et durable du niveau des prix des biens et des services. Contrairement à l'inflation, la déflation signifie que le pouvoir d'achat de la monnaie augmente au fil du temps : avec la même quantité de monnaie, on peut acquérir davantage de biens.

La déflation peut résulter d'une contraction de la masse monétaire, d'une hausse de la productivité ou d'une baisse de la demande. Les économistes keynésiens la considèrent généralement comme néfaste, car elle inciterait les agents à reporter leurs achats, freinant ainsi l'activité économique. Les économistes de l'école autrichienne distinguent cependant la déflation monétaire (contraction artificielle du crédit) de la baisse des prix liée aux gains de productivité, cette dernière étant perçue comme bénéfique.

Bitcoin est souvent qualifié de « déflationniste » en raison de son offre plafonnée à 21 millions d'unités et de la réduction progressive de son émission via les *halvings*. Plus précisément, il est désinflationniste : son taux d'émission diminue dans le temps et convergera vers zéro.

Termes associés : **HALVING** | **INFLATION**

DEMEURAGE

✿ PROTOCOLE | EN : DEMURRAGE

Coût associé à la détention d'une monnaie. Par extension, le demeurage désigne parfois un mécanisme monétaire dans lequel la détention prolongée d'une monnaie entraîne une pénalité pour son propriétaire, par exemple, en faisant diminuer progressivement sa valeur. Le but recherché est souvent d'améliorer la vitesse de circulation de la monnaie en décourageant l'épargne de long terme.

Ce mécanisme est parfois proposé pour être implémenté sur Bitcoin dans le but de soutenir le marché des frais de transaction et de répartir plus équitablement les coûts du système. En effet, le coût de fonctionnement de Bitcoin reposera à terme uniquement sur ceux qui réalisent des transactions, et pas sur les épargnants qui profitent pourtant de la sécurité du système. Cependant, le demeurage est très largement rejeté par la communauté Bitcoin, car il pénalise les détenteurs à long terme et n'est pas en adéquation avec la caractéristique de réserve de valeur du BTC. Puisqu'il impose une perte de valeur progressive, il peut être perçu comme une atteinte au droit de posséder.

On pourrait argumenter contre le demeurage en expliquant que le coût du système actuel repose déjà inévitablement sur les épargnants, étant donné qu'ils ont dû réaliser une transaction à un moment donné pour obtenir leurs bitcoins. De plus, la sécurité d'un UTXO donné augmente beaucoup avec les premiers blocs après sa création, mais avec le temps, la sécurité additionnelle obtenue à chaque bloc devient de plus en plus marginale. Ainsi, il n'est pas nécessairement injuste de ne pas faire reposer le coût du minage sur les détenteurs à long terme.

Le demeurage a notamment été appliqué sur l'altcoin Freicoin, un fork de Bit-

coin qui a implémenté un système de taux d'intérêt négatifs.

Termes associés : **FRAIS DE TRANSACTION** | **UTXO**

DÉNI PLAUSIBLE

CONFIDENTIALITÉ | EN : PLAUSIBLE DENIABILITY

Principe selon lequel un utilisateur peut nier de manière crédible la possession ou le contrôle de certains bitcoins, car il existe plusieurs interprétations plausibles d'une situation donnée. Dans le contexte de Bitcoin, le déni plausible est un objectif recherché par de nombreuses techniques de confidentialité.

Par exemple, une transaction coinjoin offre du déni plausible car un observateur ne peut pas prouver avec certitude quel output appartient à quel participant. Si l'on demande à un utilisateur s'il possède un UTXO particulier issu d'un coinjoin, il peut nier de manière crédible, car plusieurs participants pourraient en être le propriétaire.

De même, un portefeuille caché (*hidden wallet*) protégé par une passphrase BIP-0039 offre du déni plausible. Un utilisateur sous contrainte peut révéler sa phrase de récupération sans passphrase, donnant accès à un portefeuille leurre, tout en niant l'existence du portefeuille caché associé à la passphrase secrète.

Le déni plausible est un concept clé pour la protection de la vie privée et la sécurité physique des détenteurs de bitcoins, en particulier face à des adversaires étatiques ou criminels.

Termes associés : **COINJOIN** | **HEURISTIQUE D'ANALYSE**

DEPEG

ÉCONOMIE ET RÉGULATION

Fait référence à la situation où la valeur d'un actif numérique, typiquement une reconnaissance de dette, se désolidarise de l'actif sous-jacent auquel il est normalement indexé, entraînant ainsi des fluctuations de prix par rapport à cet actif. Ce phénomène est particulièrement observé avec les *stablecoins*, conçus pour maintenir une parité stricte avec des monnaies fiduciaires ou d'autres actifs de référence. Le *depeg* peut aussi survenir dans le cas de représentations de bitcoins sur des *sidechains* ou des ETF, où en cas de problème, la valeur nominale peut diverger de celle du bitcoin.

Termes associés : **SIDECHAIN** | **STABLECOIN**

DÉPÔT

‹/› INFORMATIQUE | EN : REPOSITORY / REPO

Structure de données centrale utilisée dans Git où sont stockées les informations de versionnage d'un projet. Un dépôt contient l'historique complet de toutes les modifications, les branches et les tags. Chaque dépôt est une collection indépendante de fichiers et de dossiers, accompagnée d'un historique des commits, permettant la collaboration et le suivi des changements au fil du temps. Par exemple, le dépôt de Bitcoin Core est stocké sur GitHub ici : https://github.com/bitcoin/bitcoin.

Termes associés : **GIT** | **GITHUB**

DER

‹/› INFORMATIQUE

Acronyme de *Distinguished Encoding Rules*. C'est un sous-ensemble strict des règles d'encodage ASN.1 définies dans la spécification ITU-T X.690, 2002 et utilisé pour encoder n'importe quel type de données dans une séquence binaire. DER est surtout utilisé dans des domaines spécifiques, comme en cryptographie, où les données doivent être encodées de manière standard et prédictible.

Sur Bitcoin, les signatures ECDSA sont encodées au format DER. Elles se composent de deux nombres (`r`, `s`) encodés. Le format de signature se compose des éléments suivants :

```
0x30 | length | 0x02 | r-length | r | 0x02 | s-length | s
```

Avec :

- `0x30` (1 octet) : identifiant d'une structure composée ;
- `length` (1 octet) : longueur des données suivantes ;
- `0x02` (1 octet) : identifiant de donnée type `INTEGER` (nombre entier) ;
- `r-length` (1 octet) : longueur de la valeur `r` suivante (généralement 32 ou 33 octets) ;

`r` *(variable) : valeur* `r` *en tant qu'entier gros-boutiste (*big-endian*) ;

- `0x02` (1 octet) : identifiant de donnée type `INTEGER` (nombre entier) ;
- `s-length` (1 octet) : longueur de la valeur `s` suivante (généralement 32 ou 33 octets) ;

`s` *(variable) : valeur* `s` *en tant qu'entier gros-boutiste (*big-endian*).

Dans une transaction Bitcoin, un octet est ajouté à la fin d'une signature DER pour indiquer le type de SigHash utilisé.

Termes associés : **BIP-0066** | **SIGHASH FLAG**

DÉRIVATION

⊸ PORTEFEUILLE | EN : DERIVATION

Désigne le processus de génération de paires de clés enfants à partir d'une paire de clés parent (clé privée et clé publique) et d'un code de chaîne au sein d'un portefeuille déterministe et hiérarchique (HD). Ce processus permet de

segmenter des branches et d'organiser un portefeuille en différents niveaux avec de nombreuses paires de clés enfants, qui peuvent toutes être récupérées en connaissant uniquement les informations de récupération de base (la phrase mnémonique et l'éventuelle passphrase). Pour dériver une clé enfant, on utilise la fonction `HMAC-SHA512` avec le code de chaîne parent et la concaténation de la clé parent et d'un index de 32 bits. Il existe deux types de dérivations :

- La dérivation normale, qui utilise la clé publique parent à la base de la fonction `HMAC-SHA512` ;
- La dérivation endurcie, qui utilise la clé privée parent à la base de la fonction `HMAC-SHA512` ;

Le résultat de HMAC-SHA512 est divisé en deux : les premiers 256 bits sont additionnés à la clé parent pour produire la clé enfant (privée, ou publique après multiplication par le point générateur sur la courbe elliptique), et les 256 bits restants deviennent le code de chaîne enfant.

Terme associé : **INDEX - KEY**

DÉRIVATION ENDURCIE

PORTEFEUILLE | EN : HARDENED DERIVATION

Processus de génération de clés enfants dans les portefeuilles HD. La dérivation endurcie utilise la clé privée parent comme entrée pour la fonction `HMAC-SHA512`, ce qui rend impossible la génération de clés publiques enfants à partir de la clé publique parent et du code de chaîne parent. Le processus implique la concaténation de la clé privée parent et d'un index supérieur ou égal à 2^{31}, suivi de l'application de `HMAC-SHA512` avec le code de chaîne parent. Le résultat est divisé en deux parties : les premiers 256 bits sont additionnés à la clé privée parent pour obtenir la clé privée enfant, tandis que les 256 bits restants forment le code de chaîne enfant. Cette méthode garantit que même si une clé publique étendue est compromise, elle ne peut pas être utilisée pour dériver les clés publiques enfants. Dans une dérivation standard, on utilise la dérivation endurcie à tous les niveaux de dérivation jusqu'à la profondeur des comptes. Dans les notations de chemins de dérivation, on identifie une dérivation endurcie avec une apostrophe `'` ou plus rarement avec un `h`.

Terme associé : **CHEMIN DE DÉRIVATION**

DÉRIVATION NORMALE

PORTEFEUILLE | EN : NORMAL DERIVATION

Processus de génération de clés enfants dans les portefeuilles HD. La dérivation normale utilise la clé publique parent comme entrée pour la fonction `HMAC-SHA512`, ce qui rend possible la génération de clés publiques enfants à partir de la clé publique parent et du code de chaîne parent. Le processus implique la concaténation de la clé publique parent et d'un index inférieur à 2^{31}, suivi de l'application de `HMAC-SHA512` avec le code de chaîne parent. Le résultat est divisé en deux parties : les premiers 256 bits sont additionnés à la clé privée parent pour obtenir la clé privée enfant, tandis que les 256 bits restants forment le code de chaîne enfant. Cette méthode garantit que la clé publique étendue

peut être utilisée pour dériver les clés publiques enfants. Dans une dérivation standard, on utilise la dérivation normale à tous les niveaux de dérivation à partir de la profondeur des comptes. Dans les notations de chemins de dérivation, on identifie une dérivation normale lorsqu'il y a juste l'index sans aucune apostrophe `'`.

Terme associé : **CHEMIN DE DÉRIVATION**

DGM

↗ MINAGE

Sigle de « *Double Geometric Method* ». C'est une méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. DGM est une méthode hybride, censée combiner les avantages de PPLNS et de la méthode dite « géométrique ». Elle dispose d'une faible variance sur les parts, à la manière de PPLNS, puis permet à l'opérateur de la pool d'absorber une partie du risque de variance dans un second temps. DGM est résistant au *pool hopping* en garantissant que le paiement attendu par share reste constant. La méthode est établie sur des scores, rendant les paiements indépendants de l'historique de la pool et presque totalement indépendants des changements futurs de difficulté.

Terme associé : **SHARES**

DIFFICULTÉ

↗ MINAGE | EN : DIFFICULTY

Paramètre ajustable qui détermine la complexité de la preuve de travail nécessaire pour ajouter un nouveau bloc à la blockchain et gagner la récompense associée. Cette difficulté est représentée par la cible de difficulté, une valeur de 256 bits qui fixe la limite supérieure que doit respecter le hachage de l'entête d'un bloc pour être considéré comme valide. Le but est que le hachage, réalisé via une double exécution de `SHA256` (`SHA256d`), soit inférieur ou égal à cette cible. Pour atteindre ce hachage, les mineurs manipulent le `nonce` dans l'entête du bloc. La difficulté s'ajuste tous les 2016 blocs, soit environ toutes les deux semaines, pour maintenir le temps de création de bloc moyen à environ 10 minutes.

Termes associés : **AJUSTEMENT DE LA DIFFICULTE** | **CIBLE DE DIFFICULTÉ**

DIFFIE-HELLMAN

🔒 CRYPTOGRAPHIE

Méthode cryptographique permettant à deux parties de générer un secret partagé sur un canal de communication non sécurisé. Ce secret peut ensuite servir à chiffrer la communication entre les deux parties. Diffie-Hellman utilise l'arithmétique modulaire pour que, même si un attaquant peut observer les échanges, il ne peut pas déduire le secret partagé sans résoudre un problème mathématique difficile : le logarithme discret. Sur Bitcoin, on utilise parfois une variante de DH utilisant une courbe elliptique nommée ECDH, notamment pour les protocoles d'adresse statiques comme les *Silent Payments* ou le BIP-0047.

Termes associés : **DLP - DISCRETE LOGARITHM PROBLEM** | **ECDH**

DIFFUSION

RÉSEAU | EN : PROPAGATION / BROADCAST

Processus par lequel les informations, comme les transactions et les blocs, sont transmises de nœud en nœud à travers le réseau Bitcoin. Lorsqu'un utilisateur effectue une transaction, celle-ci est d'abord vérifiée par le nœud auquel il est connecté. Après validation, cette transaction est relayée aux autres nœuds connectés à celui-ci, qui à leur tour la vérifient puis la propagent. Rapidement, une grande partie des nœuds du réseau seront en connaissance de la transaction. Si elle offre suffisamment de frais, les mineurs l'incluront dans leur bloc candidat. Lorsqu'un bloc contenant cette transaction est validé, celle-ci est alors confirmée.

Termes associés : **MEMPOOL** | **NOEUD**

DIGICASH

HISTOIRE

Entreprise fondée en 1989 par le cryptographe David Chaum, pionnière dans le domaine des paiements électroniques anonymes. DigiCash a développé le système eCash, un protocole de monnaie numérique reposant sur les signatures aveugles inventées par Chaum. Ce mécanisme permettait à une banque d'émettre des jetons numériques sans pouvoir tracer les transactions effectuées par les utilisateurs, garantissant ainsi un anonymat comparable à celui de l'argent liquide. Malgré des essais avec plusieurs banques, notamment la Mark Twain Bank aux États-Unis et la Deutsche Bank, le système n'a pas réussi à atteindre une adoption suffisante. DigiCash a fait faillite en 1998. Le concept d'eCash est considéré comme un précurseur important de Bitcoin, notamment pour son approche de la confidentialité dans les paiements numériques.

Terme associé : **SIGNATURE AVEUGLE**

DIGITAL ARTIFACTS

COUCHE SUPÉRIEURE | FR : ARTEFACT NUMÉRIQUE

Dans le contexte du protocole Ordinals, c'est un sat qui a été inscrit avec des données spécifiques via le mécanisme d'inscriptions. Ces artefacts peuvent inclure des images, des textes, ou tout autre type de contenu numérique et sont liés indissociablement au satoshi correspondant.

Termes associés : **INSCRIPTIONS** | **ORDINALS THEORY**

DIRECTED ACYCLIC GRAPH

RGB | FR : GRAPHE ORIENTÉ ACYCLIQUE

Un DAG est une structure mathématique composée de nœuds interconnectés par des arcs orientés, sans former de cycle, ce qui permet un ordonnancement topologique rigoureux. Dans le cadre du protocole RGB, le DAG sert à représenter la structure des *Shards* de contrats, pour la gestion et la vérification des transitions d'état. Cette représentation graphique permet d'assurer la cohérence et la traçabilité des modifications dans un environnement distribué.

Termes associés : **RGB** | **STATE TRANSITION**

DISTRIBUÉ

RÉSEAU | EN : DISTRIBUTED

Attribut d'un réseau informatique dans lequel le pouvoir de décision et le contrôle sont répartis de manière équitable entre tous les participants du réseau. Cette répartition garantit la résilience du système. On parle également de réseau pair-à-pair. Contrairement à un réseau décentralisé, où le pouvoir est fragmenté et dispersé parmi plusieurs entités, mais où certaines autorités centrales demeurent dotées d'un pouvoir supérieur à celui des utilisateurs, un réseau distribué élimine l'autorité centrale en confiant la gestion et le contrôle aux utilisateurs eux-mêmes. Bitcoin est un exemple de système distribué. Bitcoin ne dispose pas de hiérarchie ou d'autorité centrale. La tenue du consensus, la vérification des transactions et l'émission de nouvelles unités monétaires sont réalisées par les utilisateurs. Cette structure distribuée assure la résilience et la résistance à la censure du système, rendant très difficile pour une entité unique de le contrôler ou de le manipuler.

Certaines personnes parlent de Bitcoin comme d'un système décentralisé. En effet, il n'est pas rare d'observer une interchangeabilité de ces deux termes. Un synonyme plus évocateur de l'adjectif « distribué » pourrait être « pair-à-pair », parfois abrégé « P2P », le sigle de la traduction anglaise « Peer-to-Peer ».

Termes associés : **CONSENSUS** | **NOEUD**

DKG

CRYPTOGRAPHIE | FR : GENERATION DISTRIBUEE DE CLES

Type de protocole qui permet à un ensemble de n participants de générer collectivement une paire de clés cryptographiques, sans recourir à un tiers de confiance et sans qu'aucun participant ne connaisse la clé privée complète. Chaque participant obtient une part de la clé privée, tandis que la clé publique commune est accessible à tous. L'acronyme « DKG » signifie *Distributed Key Generation*.

Termes associés : **FROST** | **MULTISIG** | **SHAMIR SECRET SHARING - SSS**

DLC - DISCREET LOG CONTRACT

COUCHE SUPÉRIEURE

Type de contrat intelligent sur Bitcoin qui permet l'exécution de conditions contractuelles à partir du résultat d'événements externes, validés par un ou plusieurs oracles, sans que ces derniers connaissent les détails du contrat. Les DLC ont été inventés par Tadge Dryja en 2017. Ces contrats intelligents sont principalement utiles dans des applications financières, permettant, par exemple, de créer des instruments financiers ou des paris conditionnels, tout en réduisant les risques de contrepartie. Pour construire un DLC, plusieurs parties bloquent des bitcoins sur une adresse multisig. Ces bitcoins ne peuvent être débloqués que lorsque l'oracle publie les informations spécifiées à un moment donné.

Termes associés : **CONTRAT INTELLIGENT** | **ORACLE**

DLP - DISCRETE LOGARITHM PROBLEM

CRYPTOGRAPHIE | FR : PROBLÈME DU LOGARITHME DISCRET

Le problème du logarithme discret (DLP) est un problème mathématique sur lequel s'appuie la sécurité des algorithmes cryptographiques à clé publique, notamment ceux utilisés sur Bitcoin. Dans un groupe cyclique d'ordre q, avec un générateur g, si l'on a une équation de la forme $g^x = h$, alors x est appelé le logarithme discret de h par rapport à la base g, modulo q. En termes simples, il s'agit de déterminer l'exposant x lorsqu'on connaît g, h, et q. Le logarithme discret est donc la réciproque de l'exponentielle dans un groupe cyclique fini. Cependant, pour de grandes valeurs de q, résoudre le problème du logarithme discret est considéré comme algorithmiquement difficile. Cette propriété est exploitée pour assurer la sécurité de nombreux protocoles cryptographiques, tels que le protocole de Diffie-Hellman pour l'échange de clés. Le logarithme discret est aussi utilisé dans la cryptographie à courbes elliptiques (ECC), entre autres dans l'algorithme ECDSA (*Elliptic Curve Digital Signature Algorithm*). Dans le contexte des courbes elliptiques, le problème du logarithme discret s'étend à la recherche d'un scalaire k tel que $k \cdot G = K$, où G et K sont des points sur la courbe, et $\cdot$ représente l'opération de multiplication de points. Dans le contexte de Bitcoin, les transactions standards utilisent soit ECDSA, soit le protocole de Schnorr, afin de bloquer des UTXOs. Ils reposent tous deux sur l'impossibilité de calculer le logarithme discret.

Termes associés : **DIFFIE-HELLMAN** | **ECDSA**

DNS SEEDS

RÉSEAU

Points de connexion initiaux pour les nouveaux nœuds Bitcoin qui rejoignent le réseau. Ces seeds, qui sont en fait des serveurs DNS spécifiques, ont leur adresse intégrée de façon permanente dans le code de Bitcoin Core. Lorsqu'un nouveau nœud se lance, il contacte ces serveurs pour obtenir une liste aléatoire d'adresses IP de nœuds Bitcoin a priori actifs. Le nouveau nœud pourra ainsi établir des connexions avec les nœuds de cette liste afin d'obtenir les informations pour faire son IBD et se synchroniser sur la chaîne avec le plus de travail accumulé. En 2024, voici la liste des DNS seeds de Bitcoin Core et les personnes responsables de leur maintenance (https://github.com/bitcoin/bitcoin/blob/master/src/kernel/chainparams.cpp) :

- seed.bitcoin.sipa.be : Pieter Wuille ;
- dnsseed.bluematt.me : Matt Corallo ;
- dnsseed.bitcoin.dashjr-list-of-p2p-nodes.us : Luke Dashjr ;
- seed.bitcoin.jonasschnelli.ch : Jonas Schnelli ;
- seed.btc.petertodd.net : Peter Todd ;
- seed.bitcoin.sprovoost.nl : Sjors Provoost ;
- dnsseed.emzy.de : Stephan Oeste ;
- seed.bitcoin.wiz.biz : Jason Maurice ;
- seed.mainnet.achownodes.xyz : Ava Chow.

Les DNS seeds représentent le second moyen, par ordre de priorité, pour un

nœud Bitcoin d'établir des connexions. Le premier moyen consiste à utiliser le fichier `peers.dat` que le nœud a lui-même créé. Ce fichier est naturellement vide dans le cas d'un nouveau nœud, à moins que l'utilisateur l'ait modifié manuellement.

Attention, les DNS seeds ne doivent pas être confondus avec les « seed nodes », qui sont eux la troisième manière d'établir des connexions.

Terme associé : **SEED NODES**

DOJO

OUTIL

Serveur *backend open source* construit au-dessus de Bitcoin Core, orienté confidentialité et autonomie. Initialement développé par les équipes de Samourai Wallet, le projet Dojo est dorénavant maintenu par la communauté. Dojo sert de passerelle entre un portefeuille Bitcoin (comme Ashigaru) et le réseau Bitcoin, en permettant d'interroger les données de la blockchain et de diffuser des transactions via sa propre infrastructure. Il inclut Bitcoin Core, l'indexeur Fulcrum et l'explorateur BTC-RPC Explorer.

Termes associés : **ASHIGARU** **BITCOIN CORE**

DOS - DENIAL OF SERVICE

ATTAQUE | FR : ATTAQUE PAR DÉNI DE SERVICE

Attaque informatique qui vise à rendre une ressource (site web, nœud, service en ligne...) indisponible pour ses utilisateurs légitimes. Les attaquants surchargent la cible avec un volume de données ou de requêtes excessivement élevé, ce qui épuise les ressources système et réseau de la victime, entraînant des ralentissements ou un arrêt complet. Les méthodes de DoS peuvent varier, mais l'objectif reste le même : empêcher l'accès à des services ou des données. Dans le contexte spécifique de Bitcoin, une attaque DoS peut viser à saturer le réseau ou les nœuds avec un volume excessif de requêtes afin d'entraver leur fonctionnement normal. L'objectif est souvent de nuire à un opérateur de nœud ou à la disponibilité du réseau pour les utilisateurs honnêtes.

Termes associés : **ATTAQUE DES 51%** **NOEUD**

DOUBLE DÉPENSE

ATTAQUE | EN : DOUBLE SPENDING ATTACK

Attaque où un utilisateur malveillant tente d'utiliser le même UTXO (*Unspent Transaction Output*) plus d'une fois afin de s'enrichir sur les contreparties des transactions impliquées. En principe, une fois qu'une transaction est confirmée dans un bloc et ajoutée à la blockchain, l'utilisation de ces bitcoins est enregistrée de manière permanente, empêchant toute dépense ultérieure de ces mêmes bitcoins. Prévenir la double dépense est même l'utilité première de la blockchain.

Dans le cadre d'une attaque de double dépense, l'attaquant effectue d'abord une transaction légitime auprès d'un commerçant, puis crée une seconde transaction concurrente qui dépense les mêmes pièces, soit en les renvoyant vers

lui-même pour récupérer la somme, soit en les utilisant pour acheter un autre bien ou service auprès d'un autre commerçant.

Deux scénarios principaux peuvent permettre cette attaque. Le premier, et le plus simple pour l'attaquant, consiste à exécuter la transaction frauduleuse avant que la transaction légitime ne soit incluse dans un bloc. Pour permettre la confirmation de sa transaction frauduleuse en première, l'attaquant y associe des frais de transaction nettement plus élevés que la transaction légitime. C'est une sorte de RBF frauduleux. Ce scénario n'est possible que si le commerçant accepte de finaliser la vente en « zeroconf », c'est-à-dire sans aucune confirmation pour la transaction de paiement. C'est pourquoi il est fortement recommandé d'attendre plusieurs confirmations avant de considérer une transaction comme immuable. Le second scénario, beaucoup plus complexe, est celui d'une attaque à 51 %. Si l'attaquant contrôle une part importante de la puissance de calcul du réseau, il peut miner une chaîne concurrente à celle contenant la transaction légitime, mais incluant sa transaction frauduleuse. Lorsque le commerçant a accepté la vente et que l'attaquant a réussi à créer une chaîne plus longue (avec plus de travail accumulé) que la chaîne légitime, il peut alors diffuser sa chaîne frauduleuse qui sera reconnue par les nœuds du réseau comme étant celle valide.

Termes associés : **ATTAQUE DES 51%** | **ZEROCONF**

DOXXIC CHANGE

CONFIDENTIALITÉ

Output de change issu d'une transaction *Tx0* dans le protocole *Whirlpool*, qui représente un risque pour la confidentialité de l'utilisateur. Le *doxxic change* est la fraction de bitcoins qui ne peut pas être intégrée dans la pool de coinjoin choisie et qui retourne dans le portefeuille sous forme de change.

Ce change est qualifié de « doxxic » (mot-valise de « *toxic* » et « *doxx* », c'est-à-dire exposer l'identité de quelqu'un) car il conserve un lien direct avec la *Tx0*, et donc potentiellement avec l'identité ou l'historique de l'utilisateur. Si ce change est dépensé de manière imprudente, par exemple en le fusionnant avec des UTXOs *postmix*, il peut compromettre la confidentialité acquise lors des cycles de coinjoin.

Pour limiter ce risque, le *doxxic change* peut être dépensé séparément (par exemple en carte cadeaux, via des dons, ou sur *Lightning*), utilisé dans une nouvelle *Tx0*, ou bien consolidé via des *atomic swaps* sur Monero.

Termes associés : **CHANGE** | **TX0**

DRIVECHAIN

SIDECHAIN

Forme spécifique de sidechain où les mineurs de la blockchain principale (Bitcoin) ont un rôle direct dans la gouvernance de l'ancrage bilatéral et éventuellement dans le mécanisme de consensus de la sidechain. Ce protocole a été inventé par Paul Sztorc et pourrait être mis en place grâce aux controversés BIP-0300, qui permettrait le *two-way peg* auprès des mineurs, et BIP-0301, qui permettrait d'utiliser le minage fusionné aveugle (*Blind Merged Mining*).

Termes associés : **MINAGE FUSIONNÉ** | **SIDECHAIN**

DUAL FUNDING

LIGHTNING NETWORK | FR : FINANCEMENT DOUBLE

Lors de l'ouverture d'un canal Lightning, le *dual funding* permet aux deux participants d'y bloquer des bitcoins. Contrairement au modèle traditionnel où un seul nœud finance le canal, cette méthode permet aux deux parties de le financer, ce qui permet d'équilibrer ses liquidités immédiatement.

Le *dual funding* est inclus dans la version 2 du protocole de construction des canaux, qui utilise une phase de négociation. Les deux parties collaborent pour créer la transaction de financement en échangeant des messages pour ajouter ou retirer des bitcoins jusqu'à ce que la transaction soit finalisée.

Termes associés : **CHANNEL FACTORIES** | **LIQUIDITÉS**

DUMMY ELEMENT

PROTOCOLE | FR : ÉLÉMENT FACTICE

Fait référence à un élément supplémentaire et inutile consommé par les opcodes `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY` lors de la vérification des signatures dans une transaction. En raison d'un bug *off-by-one* historique (erreur de décalage unitaire), ces 2 opcodes suppriment un élément supplémentaire sur la pile en plus de leur fonction de base. Pour éviter une erreur, il est donc obligatoire d'inclure une valeur factice au début du `scriptSig` afin de satisfaire la suppression et outrepasser le bug. Cette valeur inutile, c'est ce que l'on appelle le « *dummy element* ». Le BIP-0011, qui a introduit le standard P2MS, conseillait de mettre un `OP_0` comme valeur inutile. Mais ce standard n'était pas imposé au niveau des règles de consensus, ce qui veut dire que n'importe quelle valeur pouvait y être placée, sans invalider la transaction. Le dummy element était donc un vecteur de malléabilité des transactions. Le BIP-0147, introduit avec le soft fork SegWit, a imposé que cet élément factice soit strictement un tableau d'octets vide (`OP_0`), éliminant ainsi la malléabilité associée à cet élément en rendant toute transaction non conforme invalide selon les règles de consensus. Cette règle, nommée `NULLDUMMY`, s'applique à la fois aux transactions SegWit et pré-SegWit.

Terme associé : **BIP-0147**

DURESS WALLET

⊸ PORTEFEUILLE	FR : PORTEFEUILLE DE CONTRAINTE

Portefeuille secondaire conçu pour offrir un déni plausible en cas de contrainte physique. Lorsqu'un utilisateur est forcé de déverrouiller son *signing device* sous la menace (scénario dit « *5 dollars wrench attack* »), il saisit un code PIN spécial dit « PIN de contrainte » qui ouvre un portefeuille qui contient une somme réduite. L'agresseur ne peut pas déterminer si un autre portefeuille existe sur l'appareil.

Le *duress wallet* dérive d'une seed ou d'un chemin de dérivation distinct de celui du portefeuille principal. Son interface et son fonctionnement sont identiques, ce qui empêche de distinguer les deux. Sur certains *signing devices* comme le Coldcard, cette fonctionnalité repose sur un PIN dédié ; sur d'autres appareils, la passphrase BIP-0039 remplit un rôle similaire en donnant accès à un portefeuille entièrement différent selon la passphrase saisie.

Le principe suppose de maintenir un solde crédible dans le *duress wallet* pour que le leurre soit convaincant.

Terme associé : **PASSPHRASE - BIP-0039**

DUST

⊸ PORTEFEUILLE	FR : POUSSIÈRE

Fait référence à des montants de pièces bitcoin extrêmement petits qui sont trop minimes pour être envoyés dans une transaction, car les frais de transaction nécessaires pour les inclure dans un bloc seraient proportionnellement plus élevés que leur valeur. La définition précise de « dust » peut varier selon le contexte, mais il s'agit généralement de toute sortie de transaction qui nécessite plus de frais pour être dépensée qu'elle n'incarne de valeur. Pour l'utilisateur de Bitcoin, il est important de gérer ses UTXOs et de pratiquer la consolidation de ceux-ci afin qu'ils ne deviennent pas du dust.

Termes associés : **CONSOLIDATION** | **DUST LIMIT**

DUST LIMIT

✿ PROTOCOLE	FR : LIMITE DE POUSSIÈRE

Désigne le seuil en sats en deçà duquel un UTXO est considéré comme de la « poussière » (*dust*) par un nœud du réseau. Ce seuil fait partie des règles de standardisation qui peuvent être modifiées indépendamment par chaque nœud. Dans Bitcoin Core, cette limite est déterminée par un taux de frais spécifique, fixé par défaut à 3 000 sats par kilo-octet virtuel (sats/kvB). Cette limite vise à restreindre la propagation de transactions comprenant de très petits montants en bitcoins. En effet, un UTXO qualifié de poussière implique que son utilisation n'est économiquement pas rationnelle : dépenser cet UTXO coûterait plus cher que de simplement l'abandonner. Si dépenser de la poussière n'est pas rationnel, cela suggère que de telles dépenses ne puissent être motivées que par des incitations externes, souvent malveillantes. Cela peut notamment poser un problème si un acteur malintentionné cherche à saturer le réseau avec des transactions contenant des montants infimes, dans le but d'accroître la charge

opérationnelle des nœuds et potentiellement les empêcher de traiter d'autres transactions légitimes. Pour donner une analogie (un peu bancale, je vous l'accorde), c'est un peu comme si quelqu'un tentait de payer un panier de courses de 100 € uniquement en pièces de 1 centime afin de bloquer les autres clients de la caisse.

Termes associés : **DUST** | **DUSTING ATTACK** | **DUSTRELAYFEE**

DUSTING ATTACK

ATTAQUE | FR : ATTAQUE DE POUSSIÈRE

Attaque qui consiste à envoyer de minuscules quantités de bitcoins à un grand nombre d'adresses de réception. L'objectif de l'attaquant est de pousser les destinataires à regrouper ces sommes avec d'autres UTXOs. L'attaquant suit ensuite les déplacements futurs de ces faibles quantités de bitcoins, dans le but de former des *clusters* d'adresses, c'est-à-dire de déterminer si plusieurs adresses appartiennent à une même entité. En croisant les informations recueillies lors d'une dusting attack avec d'autres données et heuristiques utilisées dans l'analyse de chaîne, il est possible pour l'attaquant d'identifier certaines entités et les adresses associées. Cette méthode représente une menace uniquement pour la confidentialité des utilisateurs, mais n'affecte pas la sécurité de leurs fonds.

Certains bitcoiners suggèrent de ne plus utiliser le terme de « dusting attack » car celui-ci induirait en erreur. En effet, le terme de « dust » décrit quelque chose de bien précis dans Bitcoin Core. Si la dusting attack utilisait réellement du dust comme décrit dans Core, l'attaque serait inefficace. Certains suggèrent ainsi d'utiliser le terme de « forced address reuse » (réutilisation d'adresse forcée) pour décrire plus précisément cette attaque.

Termes associés : **ANALYSE DE CHAINE** | **RÉUTILISATION D'ADRESSE**

DUSTRELAYFEE

PROTOCOLE

Règle de standardisation utilisée par les nœuds du réseau pour déterminer ce qu'ils considèrent comme la « limite de poussière » (dust limit). Ce paramètre fixe un taux de frais en sats par kilo-octet virtuel (sats/kvB) qui sert de référence pour calculer si la valeur d'un UTXO est inférieure aux frais nécessaires pour l'inclure dans une transaction. En effet, un UTXO est considéré comme « dust » (poussière) sur Bitcoin s'il requiert plus de frais pour être transféré que la valeur qu'il représente lui-même. Le calcul de cette limite est le suivant :

```
limite = (taille entrée + taille sortie) * taux de frais
```

Comme le taux de frais requis pour qu'une transaction soit incluse dans un bloc Bitcoin varie constamment, le paramètre `DustRelayFee` permet de définir le taux de frais utilisé dans ce calcul par chaque nœud. Par défaut, sur Bitcoin Core, cette valeur est fixée à 3 000 sats/kvB. Par exemple, pour calculer la limite de poussière d'une entrée et d'une sortie P2PKH, qui mesurent respectivement 148 et 34 octets, le calcul serait :

```
limite (sats) = (148 + 34) * 3000 / 1000 = 546 sats
```

Cela signifie que le nœud en question ne relayera pas les transactions incluant un UTXO sécurisé en P2PKH dont la valeur est inférieure à 546 sats.

Termes associés : **DUST** | **DUST LIMIT**

DYNAFED - DYNAMIC FEDERATIONS

SIDECHAIN | FR : FÉDÉRATIONS DYNAMIQUES

Mécanisme de gouvernance déployé sur le réseau Liquid qui permet de modifier la composition de la fédération de fonctionnaires sans interrompre le fonctionnement du réseau. Avant l'introduction des Dynamic Federations (DynaFed), tout changement dans la liste des signataires de blocs ou des gardiens (*watchmen*) nécessitait une coordination manuelle et une interruption du réseau. DynaFed a été activé sur Liquid en octobre 2021 par un *hard fork*.

Termes associés : **FÉDÉRATION** | **LIQUID NETWORK**

E

E-GOLD

HISTOIRE

Système de monnaie numérique adossée à l'or, fondé en 1996 par Douglas Jackson et Barry Downey. Les utilisateurs pouvaient ouvrir des comptes libellés en grammes de métaux précieux (or, argent, platine, palladium) stockés dans des coffres. Les transferts entre comptes étaient instantanés et irrévocables.

À son apogée en 2006, e-gold traitait plus de 2 milliards de dollars de transactions par an et comptait environ 3,5 millions de comptes. Le système a attiré à la fois des utilisateurs légitimes et des activités illicites, du fait de l'absence de vérification d'identité.

En 2007, le gouvernement américain a inculpé les opérateurs d'e-gold pour exploitation d'un service de transfert d'argent sans licence et blanchiment de capitaux. La société a plaidé coupable en 2008. La fermeture d'e-gold a illustré la vulnérabilité des systèmes de monnaie numérique centralisés face à l'action étatique, un constat qui a influencé la conception décentralisée de Bitcoin.

Termes associés : **DIGICASH** | **ECASH - DAVID CHAUM**

ECASH - DAVID CHAUM

HISTOIRE

Protocole proposé par David Chaum en 1982, qui est un des premiers systèmes de monnaie numérique conçu pour préserver l'anonymat des utilisateurs. Il repose sur des principes de cryptographie à clé publique pour créer une monnaie numérique qui peut être échangée de manière sécurisée et anonyme. eCash fonctionne par la création de jetons numériques signés par une banque. C'est donc une évolution des banques de dépôt, sans pour autant être décentralisée. Lors des transactions, ces jetons sont transférés entre les parties sans révéler l'identité des utilisateurs, préservant ainsi leur vie privée. eCash est considéré comme un précurseur des cryptomonnaies. Il revient d'ailleurs souvent dans les discussions autour de Bitcoin, certains voulant utiliser des systèmes similaires à eCash en surcouche. Aujourd'hui, la tendance est plutôt aux systèmes dits « chaumiens fédérés » comme Fedimint.

Termes associés : **FEDIMINT** | **SIGNATURE AVEUGLE**

ECASH - XEC

HISTOIRE

Système de cryptomonnaie, précédemment connu sous le nom de Bitcoin Cash ABC (BCHA), issu d'un hard fork de Bitcoin Cash (BCH). Le fork d'eCash est survenu le 15 novembre 2020 au bloc 661 648, résultant d'un conflit au sein de la communauté Bitcoin Cash.

Termes associés : **ECASH - DAVID CHAUM** | **HARD FORK**

ECDH

CRYPTOGRAPHIE

Méthode d'échange de clés cryptographiques établie sur les principes de l'échange de clés Diffie-Hellman, mais qui utilise des courbes elliptiques pour fournir un niveau de sécurité élevé avec des tailles de clés plus petites. Ce protocole permet à deux parties de générer un secret partagé en utilisant leurs paires de clés publiques et privées, sans jamais avoir à échanger les clés privées elles-mêmes. Le secret partagé peut ensuite être utilisé pour chiffrer une communication ultérieure. On retrouve parfois l'utilisation de cet algorithme dans des propositions d'amélioration de Bitcoin, notamment le BIP-0047 ou le BIP-0352 pour la dérivation d'adresses de réception vierges à partir d'un identifiant statique.

Termes associés : **BIP-0352** **DIFFIE-HELLMAN**

ECDSA

CRYPTOGRAPHIE

Sigle de « *Elliptic Curve Digital Signature Algorithm* ». C'est un algorithme de signature numérique établi sur la cryptographie à courbes elliptiques (ECC). Il s'agit d'une variante de l'algorithme DSA (*Digital Signature Algorithm*). ECDSA utilise les propriétés des courbes elliptiques pour procurer des niveaux de sécurité comparables à ceux des algorithmes à clé publique traditionnels, tels que RSA, tout en utilisant des clés de taille nettement inférieure. ECDSA permet la génération de paires de clés (clé publique et clé privée) ainsi que la création et la vérification de signatures numériques.

Dans le contexte de Bitcoin, ECDSA est utilisé pour dériver des clés publiques, à partir de clés privées. Il est également utilisé pour signer les transactions, afin de satisfaire un script pour déverrouiller des bitcoins et les dépenser. La courbe elliptique utilisée sur Bitcoin au sein d'ECDSA est `secp256k1`, définie par l'équation $y^2 = x^3 + 7$. Cet algorithme est celui utilisé dès les débuts de Bitcoin en 2009. Aujourd'hui, il partage sa place avec le protocole de Schnorr, un autre algorithme de signature électronique introduit avec Taproot en 2021.

Termes associés : **SCHNORR** **SECP256K1**

ECLAIR

LIGHTNING NETWORK

Implémentation majeure du protocole Lightning Network écrite en langage Scala. Eclair est développé par la société française ACINQ.

Attention, « Eclair » était également le nom d'un portefeuille Lightning pour les appareils mobiles, développé par la même société. Aujourd'hui, ce portefeuille n'est plus maintenu.

Termes associés : **LIGHTNING NETWORK** **LND**

ECLIPSE

ATTAQUE

Attaque qui consiste à isoler et à contrôler les communications d'un nœud dans un réseau en créant un environnement artificiel autour de lui. L'objectif est de filtrer ou de manipuler les informations reçues et envoyées par ce nœud, le coupant ainsi de ses pairs légitimes. Dans le cadre de Bitcoin, cette technique peut être utilisée pour induire en erreur un nœud, censurer ou altérer les données qu'il reçoit ou envoie, ou pour mener des attaques de doubles dépenses.

Termes associés : **DOUBLE DÉPENSE** | **NOEUD**

ÉCOLE AUTRICHIENNE

ÉCONOMIE ET RÉGULATION | EN : AUSTRIAN SCHOOL OF ECONOMICS

École de pensée économique qui théorise le marché comme un ensemble d'interactions individuelles volontaires, souligne la spontanéité de l'ordre économique et critique les interventions étatiques. L'École Autrichienne défend le rôle de la propriété privée, de la liberté contractuelle, et du libre-échange, tout en critiquant les effets perturbateurs de la création monétaire sur l'économie. Ses contributeurs, tels que Carl Menger, Ludwig von Mises ou Friedrich Hayek, ont travaillé des concepts tels que la formation des prix, la fonction de la monnaie, les dynamiques du capital ou encore la théorie subjective de la valeur. L'École Autrichienne critique le socialisme pour son incapacité à réaliser des calculs économiques efficaces, et favorise une approche libérale. Elle valorise le marché libre et voit dans l'interventionnisme étatique une source de déséquilibres économiques. De nombreux bitcoiners adhèrent à ces idées et estiment que Bitcoin représente un outil en harmonie avec cette philosophie en raison de sa nature distribuée, de la limitation de sa création monétaire et de sa capacité à fonctionner indépendamment de l'intervention étatique.

Termes associés : **AGORISME** | **INFLATION**

EDGE NODE

COUCHE SUPÉRIEURE | FR : NŒUD DE BORDURE

Nœud de routage spécialisé servant d'intermédiaire entre les portefeuilles Taproot Assets et le réseau Lightning classique. Un *edge node* maintient des canaux Bitcoin publics vers le réseau Lightning et des canaux Taproot Assets non annoncés avec ses clients. Sa fonction principale est de convertir les Taproot Assets en BTC (et inversement) lors du routage des paiements, via des HTLC standards.

Concrètement, un portefeuille détenant un actif Taproot (par exemple un stablecoin dollar) peut payer n'importe quelle facture Lightning BOLT-11/12 : l'*edge node* reçoit l'actif du client, effectue la conversion et route le paiement en BTC sur le réseau. Inversement, un paiement Lightning entrant en BTC peut être converti en Taproot Asset avant d'être livré au client. L'*edge node* fournit des cotations (*quotes*) pour chaque conversion et perçoit des frais de routage. Il ne prend jamais custody des fonds du client : les échanges sont atomiques grâce aux HTLC.

Termes associés : **HTLC** | **TAPROOT ASSETS PROTOCOL**

EFFET CANTILLON

Ⓢ ÉCONOMIE ET RÉGULATION | EN : CANTILLON EFFECT

Phénomène économique décrit par l'économiste irlando-français Richard Cantillon au XVIIIe siècle, selon lequel l'injection de nouvelle monnaie dans une économie ne se diffuse pas de manière uniforme. Les premiers bénéficiaires de cette monnaie nouvellement créée (banques, institutions financières, États) peuvent l'utiliser avant que les prix n'augmentent, tandis que les derniers à la recevoir (salariés, épargnants, populations les plus éloignées du système bancaire) subissent la hausse des prix sans avoir profité de l'accès précoce à cette liquidité. Cela crée une redistribution cachée des richesses au profit de ceux qui sont proches de la source d'émission monétaire.

Bitcoin est parfois présenté comme une réponse à cet effet : son émission monétaire est programmée, prévisible et régie par le protocole. La subvention de bloc, qui décroît au fil des halvings, est distribuée aux mineurs en contrepartie d'un travail, sans qu'aucune autorité centrale ne puisse décider de créer arbitrairement de nouvelles unités. Il n'existe donc pas d'intermédiaire privilégié capable de capter la valeur de la création monétaire avant le reste du réseau.

Termes associés : **INFLATION** | **QE - QUANTITATIVE EASING** | **ÉCOLE AUTRICHIENNE**

EFFET DE RÉSEAU

Ⓢ ÉCONOMIE ET RÉGULATION | EN : NETWORK EFFECT

Phénomène économique dans lequel la valeur ou l'utilité d'un produit ou d'un service augmente à mesure que le nombre de ses utilisateurs croît. Plus un réseau compte de participants, plus il devient attractif pour de nouveaux utilisateurs, créant un cercle vertueux d'adoption. Ce concept est étroitement lié à la loi de Metcalfe.

L'effet de réseau est fondamental pour comprendre la monnaie : une monnaie n'a de valeur que parce que d'autres personnes l'acceptent. Plus le nombre de personnes qui acceptent une monnaie est grand, plus cette monnaie est utile comme moyen d'échange. Ce mécanisme explique la tendance naturelle vers un standard monétaire unique au sein d'une économie.

Terme associé : **DÉCENTRALISATION**

ELECTRS

RÉSEAU

Implémentation *open-source* d'Electrum Server écrite en Rust. Le rôle d'un electrs est donc de maintenir un index complet des adresses et des transactions Bitcoin depuis la blockchain fournie par un nœud complet, afin de faire facilement et rapidement des requêtes depuis un logiciel de portefeuille (comme Electrum par exemple).

Terme associé : **ELECTRUM SERVER**

ELECTRUM

PORTEFEUILLE

Portefeuille Bitcoin fondé en novembre 2011 par Thomas Voegtlin, qui permet aux utilisateurs de gérer leurs fonds sans télécharger l'intégralité de la blockchain grâce à un système SPV (*Simplified Payment Verification*). Du fait de son existence depuis le tout début des années 2010 et de son implication dans le développement de Bitcoin, Electrum occupe une place historique en tant que logiciel de portefeuille.

Termes associés : **ELECTRUM SERVER** | **NOEUD SPV - NOEUD LÉGER**

ELECTRUM LIGHTNING

LIGHTNING NETWORK

Implémentation du Lightning Network écrite en Python spécifiquement pour le logiciel Electrum.

Termes associés : **ELECTRUM** | **LIGHTNING NETWORK**

ELECTRUM SERVER

RÉSEAU

Indexeur implémenté sur un nœud complet Bitcoin qui permet aux logiciels de portefeuilles Electrum (ou tout autre portefeuille les prenant en charge) de fonctionner sans télécharger l'intégralité de la blockchain de Bitcoin. Un Electrum Server maintient un index complet des adresses et des transactions Bitcoin, ce qui permet de faire des requêtes rapides depuis un autre logiciel. Pour vulgariser, un Electrum Server fait une sorte de pont entre un logiciel de portefeuille et un nœud Bitcoin qui dispose de la blockchain.

Termes associés : **ELECTRS** | **ELECTRUM**

ELEMENTS

SIDECHAIN

Plateforme blockchain open source permettant de déployer des sidechains fédérées rattachées à Bitcoin ou des blockchains autonomes. Lancé en juin 2015, Elements est construit sur le code de Bitcoin Core et l'étend avec des fonctionnalités avancées : *Confidential Transactions* (montants et types d'actifs masqués), *Issued Assets* (émission de plusieurs types d'actifs sur une même chaîne), opcodes supplémentaires et *Federated Peg* (ancrage bidirectionnel avec Bitcoin via une fédération de signataires).

Le consensus repose sur des blocs signés par une fédération de participants aux intérêts alignés, ce qui élimine le risque de réorganisations multi-blocs et accélère la finalité des transactions par rapport au minage par preuve de travail. Liquid Network, la sidechain la plus utilisée de l'écosystème Bitcoin, est la principale implémentation en production d'Elements.

Termes associés : **LIQUID NETWORK** | **SIDECHAIN**

ELTOO

LIGHTNING NETWORK

Protocole généraliste pour les secondes couches de Bitcoin qui permet de définir la manière de gérer conjointement la propriété d'un UTXO. Eltoo a été conçu par Christian Decker, Rusty Russell et Olaoluwa Osuntokun, notamment pour résoudre les problèmes associés aux mécanismes de négociation de l'état des canaux Lightning, c'est-à-dire entre l'ouverture et la fermeture. L'architecture Eltoo simplifie le processus de négociation en introduisant un système de gestion des états linéaire, remplaçant l'approche établie sur la pénalité par une méthode de mise à jour plus flexible et moins punitive. Ce protocole nécessite l'utilisation d'un nouveau type de SigHash qui permette de ne prendre en compte aucune entrée dans la signature d'une transaction. Ce SigHash a d'abord été appelé `SIGHASH_NOINPUT`, puis `SIGHASH_ANYPREVOUT` (*Any Previous Output*). Son implémentation est prévue dans le BIP-0118.

Eltoo est parfois également appelé « LN-Symmetry ».

Termes associés : **CANAL DE PAIMENT** | **SIGHASH_ANYPREVOUT**

EMBRANCHEMENT NATUREL

PROTOCOLE | EN : NATURAL BRANCHING / CHAINSPLIT

Séparation temporaire de la blockchain résultant de la diffusion quasi simultanée de plusieurs blocs par différents mineurs à une même hauteur. Cette situation se produit lorsque deux blocs, désignés comme A et B, sont trouvés presque simultanément, entraînant une division temporaire du réseau. Puisque chaque nœud considère comme valide le premier bloc qu'il a reçu, mais que tout le monde n'a pas reçu le même bloc en premier, une partie des nœuds suit la chaîne contenant le bloc A, tandis que l'autre suit celle avec le bloc B. Cet embranchement est résolu lorsqu'une des deux chaînes concurrentes dépasse l'autre en termes de travail accumulé. À ce moment-là, tous les nœuds du réseau s'accordent automatiquement sur la chaîne la plus longue (avec le plus de travail accumulé), un processus que l'on appelle la réorganisation ou la resynchronisation. Ces embranchements naturels sont inhérents au fonctionnement distribué de Bitcoin. Ils sont parfaitement normaux et se résolvent spontanément au bout de quelques blocs (généralement un seul). S'ils sont trop nombreux, ces embranchements peuvent tout de même être délétères, car ils entraînent un gaspillage de la puissance de calcul sur une branche qui deviendra finalement obsolète.

Termes associés : **OBSOLÈTE - BLOC** | **RESYNCHRONISATION**

EMPREINTE DE PORTEFEUILLE

PORTEFEUILLE | EN : WALLET FINGERPRINT

Ensemble de caractéristiques distinctives observables dans les transactions effectuées par un même portefeuille Bitcoin. Ces caractéristiques peuvent inclure des similitudes dans l'utilisation des types de scripts, la réutilisation d'adresses, l'ordre des UTXOs, la place des outputs de change, la signalisation de RBF (*Replace-by-Fee*), le numéro de version, le champ `nSequence` et le champ `nLockTime`.

Les empreintes de portefeuille sont utilisées par les analystes pour tracer les activités d'une entité particulière sur la blockchain en identifiant des patterns récurrents dans ses transactions. Par exemple, un utilisateur qui envoie systématiquement son change vers des adresses P2TR (`bc1p…`) crée une empreinte caractéristique qui peut être utilisée pour suivre ses transactions futures.

Comme le précise LaurentMT dans le Space Kek #19 (un podcast francophone), l'utilité des empreintes de portefeuille dans l'analyse de chaîne s'accroît de manière significative avec le temps. En effet, le nombre croissant de types de scripts et le déploiement de plus en plus progressif de ces nouvelles fonctionnalités par les logiciels de portefeuille accentuent les différences. Il arrive même que l'on puisse identifier avec exactitude le logiciel employé par l'entité tracée. Il faut donc comprendre que l'étude de l'empreinte d'un portefeuille s'avère particulièrement pertinente pour les transactions récentes, davantage que pour celles initiées au début des années 2010.

Termes associés : **ANALYSE DE CHAINE** | **HEURISTIQUE D'ANALYSE**

ENDIANNESS

INFORMATIQUE | FR : BOUTISME

Désigne l'ordre dans lequel une séquence d'octets est arrangée et interprétée en informatique. On distingue deux types : « *big-endian* », où l'octet de poids le plus fort (le plus significatif) est stocké en premier, et « *little-endian* », où l'octet de poids le plus faible (le moins significatif) est stocké en premier.

Termes associés : **BIG-ENDIAN** | **LITTLE-ENDIAN**

ENGRAVING

RGB

Dans le cadre du protocole RGB, l'*engraving* désigne une chaîne de données optionnelle que peuvent inscrire successivement les détenteurs d'un contrat dans l'historique de ses transitions. Accessible notamment via l'interface RGB21, cette fonctionnalité permet d'ajouter des informations commémoratives ou descriptives sans modifier la validité des opérations. Elle enrichit l'historique du contrat en intégrant des métadonnées supplémentaires.

Termes associés : **CONTRAT RGB** | **STATE TRANSITION**

ENTÊTE DE BLOC

PROTOCOLE | EN : BLOCK HEADER

L'entête de bloc est une structure de données servant de composant principal dans la construction d'un bloc Bitcoin. Chaque bloc est composé d'un entête et d'une liste de transactions. L'entête de bloc contient les informations cruciales qui permettent d'assurer l'intégrité et la validité d'un bloc au sein de la blockchain. L'entête de bloc contient 80 octets de métadonnées et se compose des éléments suivants :

- La version du bloc ;
- L'empreinte du bloc précédent ;
- La racine de l'arbre de Merkle des transactions ;
- L'horodatage du bloc ;
- La cible de difficulté ;
- Le nonce.

Par exemple, voici l'entête du bloc n° 785 530 au format hexadécimal, miné par Foundry USA le 15 avril 2023 :

```
00e0ff3f5ffe3b0d9247dc437e18edc19252e4517cee941752d5010000000000
00000000206bde3a10826e2acb2f28fba70463601c789293d0c9c4348d7a0d06
711e97c0bcb13a64b2e0051743f09a40
```

Si l'on décompose cet entête, on peut reconnaitre :

- La version :

```
00e0ff3f
```

- L'empreinte précédente :

```
5ffe3b0d9247dc437e18edc19252e4517cee941752d501000000000000000000
```

- La racine de Merkle :

```
206bde3a10826e2acb2f28fba70463601c789293d0c9c4348d7a0d06711e97c0
```

- L'horodatage :

```
bcb13a64
```

- La cible :

```
b2e00517
```

- Le nonce :

```
43f09a40
```

Pour être valide, un bloc doit disposer d'un entête qui, une fois haché avec `SHA256d`, produit une empreinte inférieure ou égale à la cible de difficulté.

Termes associés : ARBRE DE MERKLE | NONCE

ENTROPIE

CRYPTOGRAPHIE | EN : ENTROPY

L'entropie, dans le contexte de la cryptographie et de l'information, est une mesure quantitative de l'incertitude ou de l'imprévisibilité associée à une source de données ou à un processus aléatoire. L'entropie joue un rôle crucial dans la sécurité des systèmes cryptographiques, notamment dans la génération de clés et de nombres aléatoires. Une entropie élevée garantit que les clés générées sont suffisamment imprévisibles et résistantes aux attaques par force brute, où un attaquant essaie toutes les combinaisons possibles pour deviner la clé.

Dans le contexte de Bitcoin, l'entropie est utilisée pour générer des clés privées ou des graines. Lors de la création d'un portefeuille déterministe et hiérarchique, la construction de la phrase mnémonique se fait à partir d'un nombre aléatoire, lui-même issu d'une source d'entropie. La phrase est ensuite utilisée pour générer plusieurs clés privées, de manière déterministe et hiérarchique, afin de créer des conditions de dépense sur des UTXOs.

Il est essentiel de disposer d'une source d'entropie de qualité pour garantir la sécurité des systèmes cryptographiques. Les sources d'entropie peuvent être des processus physiques, tels que le bruit électronique ou les variations thermiques, ou des processus logiciels, tels que les générateurs de nombres pseudo-aléatoires.

Termes associés : **GRAINE** | **PHRASE DE RÉCUPÉRATION**

ENTROPIE - ANALYSE DE CHAINE

CONFIDENTIALITÉ | EN : ENTROPY - CHAIN ANALYSIS

Dans le contexte spécifique de l'analyse de chaîne, l'entropie est également le nom d'un indicateur, dérivé de l'entropie de Shannon, inventé par LaurentMT. Cet indicateur permet de mesurer le manque de connaissance des analystes sur la configuration exacte d'une transaction Bitcoin. Autrement dit, plus l'entropie d'une transaction est élevée, plus la tâche d'identification des mouvements de bitcoins entre les entrées et les sorties devient difficile pour les analystes.

En pratique, l'entropie révèle si, du regard d'un observateur externe, une transaction présente de multiples interprétations possibles, basées uniquement sur les montants des entrées et sorties, sans prendre en compte d'autres *patterns* et heuristiques externes ou internes. Une grande entropie est alors synonyme d'une meilleure confidentialité pour la transaction.

L'entropie est définie comme le logarithme binaire du nombre de combinaisons possibles. Voici la formule utilisée avec E l'entropie de la transaction et C le nombre d'interprétations possibles :

$$E = \log_2(C)$$

En prenant en compte les valeurs des UTXOs impliqués dans la transaction, le nombre d'interprétations C représente le nombre de manières dont les entrées peuvent être associées aux sorties. Autrement dit, il détermine le nombre d'interprétations qu'une transaction peut susciter du point de vue d'un observateur

extérieur qui l'analyse.

Termes associés : **ANALYSE DE CHAINE** | **COINJOIN**

ENVOY

⊸ PORTEFEUILLE

Application mobile développée par Foundation Devices. Envoy sert à la fois de portefeuille Bitcoin autonome et d'interface compagnon pour le *hardware wallet* Passport : elle permet de configurer l'appareil, de mettre à jour son firmware et de signer des transactions via des QR codes échangés avec le Passport, qui fonctionne en *air-gap*.

Termes associés : **FOUNDATION DEVICES** | **PASSPORT**

EPHEMERAL ANCHORS

✪ PROTOCOLE | FR : SORTIES D'ANCRAGE ÉPHÉMÈRES

Proposition de politique de relais qui autorise la diffusion de transactions à frais nuls sur le réseau, à condition qu'elles soient accompagnées d'une transaction enfant qui paie des frais suffisants pour les deux. La transaction parente contient une sortie de valeur nulle avec un script *anyone-can-spend*, ce qui permet à n'importe quel participant du réseau de créer une transaction enfant qui dépense cette sortie éphémère en y attachant les frais appropriés au moment de la diffusion.

Ce mécanisme résout un problème fondamental des protocoles de contrat comme le Lightning Network : les transactions sont signées par les participants bien avant leur diffusion, ce qui empêche de déterminer à l'avance un taux de frais adapté aux conditions du réseau. Les *ephemeral anchors* permettent de reporter le choix des frais au moment de la publication, en offrant une forme de « parrainage de frais » (*fee sponsorship*) accessible à toute partie, qu'elle soit ou non bénéficiaire des autres sorties de la transaction.

Cette proposition s'appuie sur les règles de relais des transactions v3 et sur le *package relay*. Elle constitue une évolution des *anchor outputs* traditionnels du Lightning Network, eux-mêmes fondés sur la règle de *CPFP carve-out*. Bitcoin Core a introduit le type de sortie standardisé *Pay-To-Anchor* (P2A) pour implémenter ce mécanisme. Les *ephemeral anchors* constituent également une contre-mesure efficace face aux attaques d'épinglage de transaction.

Termes associés : **ANCHOR OUTPUTS** | **CPFP - CHILD PAY FOR PARENT** | **GOSSIP**

EREBUS

ATTAQUE

Forme très sophistiquée d'attaque contre le réseau Bitcoin qui permet à un fournisseur de services Internet malveillant d'isoler des nœuds Bitcoin spécifiques. C'est donc une forme d'attaque Eclipse. L'attaque Erebus exploite la structure du réseau Internet, en particulier les points de passage obligés (ou « *bottlenecks* ») dans le routage entre les systèmes autonomes (AS). Un attaquant, en contrôlant un système autonome, peut manipuler le trafic réseau pour isoler un nœud Bitcoin du reste du réseau, et ainsi lui faire croire à un faux état de la blockchain (blocs ou transactions non connues par le nœud). Cette isolation peut conduire à des doubles dépenses ou de la censure à l'encontre du nœud isolé. Cette attaque est rendue beaucoup plus difficile depuis la version 0.20.0 de Bitcoin Core et l'introduction d'Asmap.

Terme associé : **ASMAP**

ERLAY

RÉSEAU

Proposition de protocole réseau dont l'objectif est d'améliorer l'efficacité du relais des transactions non confirmées entre les nœuds Bitcoin.

Actuellement, chaque transaction est propagée via un système où chaque nœud diffuse la transaction dont il prend connaissance à l'ensemble de ses pairs. Le problème, c'est que cela entraîne de nombreuses redondances et une utilisation de la bande passante pour des doublons. De nombreuses diffusions de transactions s'avèrent inutiles, car le destinataire est déjà en connaissance de ces transactions et chaque nœud n'a besoin de connaître chaque transaction qu'une seule fois. Erlay propose de limiter par défaut à 8 le nombre de pairs auxquels un nœud envoie directement les transactions dont il prend connaissance, puis d'utiliser un processus de réconciliation basé sur la bibliothèque *minisketch* pour partager les transactions manquantes de manière plus efficace.

Erlay réduirait la consommation de bande passante d'environ 40 %, ce qui rendrait l'opération d'un nœud complet plus accessible aux utilisateurs avec des connexions Internet limitées, et ce qui favoriserait donc une meilleure décentralisation du réseau. Ce protocole maintiendrait aussi une consommation de bande passante quasi constante avec une augmentation du nombre de connexions. Cela signifie qu'il serait bien plus simple pour les opérateurs de nœuds d'accepter un très grand nombre de connexions de leurs pairs, ce qui renforcerait la sécurité du réseau Bitcoin en réduisant les risques de partitionnement, intentionnels ou accidentels. De plus, Erlay compliquerait la tâche de déterminer le nœud d'origine d'une transaction, ce qui augmenterait ainsi la confidentialité pour les utilisateurs de nœuds qui n'opèrent pas sous Tor.

Erlay est proposé dans le BIP-0330.

Termes associés : **BIP-0330** **COMPACT BLOCK RELAY**

ESCROW

$ ÉCONOMIE ET RÉGULATION | FR : SÉQUESTRE

Mécanisme par lequel des fonds sont confiés à un tiers de confiance qui les conserve jusqu'à ce que les conditions convenues entre les parties soient remplies. En français, on parle de « séquestre ». L'escrow est couramment utilisé dans les transactions commerciales pour protéger l'acheteur et le vendeur.

Dans le contexte de Bitcoin, l'escrow peut être implémenté de différentes manières. La méthode la plus courante utilise des adresses multisig (2-de-3), où trois clés sont réparties entre l'acheteur, le vendeur et un arbitre. Les fonds ne peuvent être débloqués que si au moins deux des trois parties signent la transaction, ce qui permet une résolution de conflits sans qu'aucune partie seule ne contrôle les fonds.

Des timelocks peuvent également être intégrés pour garantir que les fonds ne restent pas bloqués indéfiniment. L'escrow on-chain est notamment utilisé par les plateformes d'échange pair-à-pair comme Bisq ou HodlHodl pour sécuriser les transactions sans tiers centralisé.

Termes associés : **MULTISIG** **TIMELOCK**

ESMPPS

↗ MINAGE

Sigle de « *Equalized Shared Maximum Pay Per Share* ». C'est une méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. ESMPPS vise à répartir la récompense de manière équitable entre toutes les shares soumises, indépendamment du moment de leur soumission ou de la luck de la pool. Cela fonctionne essentiellement comme SMPPS, mais avec une notion d'égalité pour les parts soumises en plus.

Terme associé : **SHARES**

ESPLORA

✱ OUTIL

Explorateur de blocs *open source* développé par Blockstream et accessible publiquement sur `blockstream.info`. L'interface se présente sous la forme d'une application web monopage (*SPA*) qui consomme l'API HTTP fournie par *esplora-electrs*, une version modifiée d'*Electrs* en Rust qui indexe la blockchain à partir d'un nœud Bitcoin Core. L'outil supporte également Liquid et les chaînes basées sur Elements, avec la gestion des transactions confidentielles et des opérations de *peg-in/peg-out*.

Termes associés : **BLOCKSTREAM** **ELECTRS**

ETF

ÉCONOMIE ET RÉGULATION | FR : FONDS NÉGOCIÉ EN BOURSE

Un ETF (*Exchange-Traded Fund*) est un fonds d'investissement coté en bourse qui réplique la performance d'un actif ou d'un panier d'actifs. À la différence d'un fonds classique, un ETF s'achète et se vend en continu sur les marchés boursiers, comme une action ordinaire.

Dans le contexte de Bitcoin, les ETF spot Bitcoin, approuvés aux États-Unis en janvier 2024, permettent aux investisseurs institutionnels et particuliers de s'exposer au cours du bitcoin sans détenir directement l'actif. Le fonds achète et conserve des bitcoins réels pour le compte de ses porteurs de parts. Il existe aussi des ETF à terme (*futures*) qui se basent sur des contrats dérivés plutôt que sur la détention physique.

Les ETF Bitcoin impliquent un intermédiaire de confiance (le gestionnaire du fonds), des frais et ne confèrent pas la souveraineté individuelle que procure la détention directe en self-custody.

Terme associé : **SELF-CUSTODY**

ÉTIQUETAGE

PORTEFEUILLE | EN : LABELING

Pratique qui consiste à attribuer une annotation ou une étiquette à un UTXO spécifique dans un portefeuille Bitcoin. Par exemple, si je possède un UTXO provenant d'un achat P2P sur Bisq avec Charles, je pourrais lui attribuer l'étiquette « `Non-KYC Bisq Charles` ».

L'étiquetage est une bonne pratique qui aide à se rappeler l'origine ou la destination prévue d'un UTXO, ce qui facilite ainsi la gestion des fonds et l'optimisation de la confidentialité. L'étiquetage est d'autant plus important lorsqu'il est utilisé avec le *coin control*. En effet, en permettant aux utilisateurs de différencier et de sélectionner précisément les UTXOs pour leurs transactions, cette pratique aide à éviter la fusion d'UTXOs provenant de sources différentes. Cela limite les risques associés à l'heuristique d'analyse de chaîne CIOH (*Common Input Ownership Heuristic*), qui peut révéler la propriété commune des entrées d'une transaction.

Terme associé : **COIN CONTROL**

ÉVÉNEMENT NOSTR

RÉSEAU | EN : NOSTR EVENT

Unité fondamentale de données dans le protocole Nostr. Toute information transmise sur le réseau Nostr (publication de texte, mise à jour de profil, réaction, message chiffré, reçu de zap...) prend la forme d'un événement. Un événement est un objet JSON contenant sept champs :

- `id` : un identifiant unique obtenu par hachage SHA-256 du contenu sérialisé,
- `pubkey` : la clé publique de l'auteur,
- `created_at` : l'horodatage Unix,
- `kind` : le type d'événement identifié par un entier,
- `tags` : les métadonnées structurées permettant de mentionner des utilisa-

teurs, référencer d'autres événements ou ajouter des hashtags,
- `content` : le contenu textuel,
- `sig` : la signature Schnorr sur secp256k1.

Le champ `kind` détermine la nature de l'événement : kind 0 pour les métadonnées de profil, kind 1 pour les notes textuelles, kind 4 pour les messages directs chiffrés, kind 7 pour les réactions ou kind 9735 pour les reçus de zap. Les relais vérifient la signature avant d'accepter un événement, ce qui garantit l'authenticité et l'intégrité des données.

Termes associés : **NOSTR** | **SHA256**

EXIT TRANSACTION

COUCHE SUPÉRIEURE | FR : TRANSACTION DE SORTIE

Transaction Bitcoin pré-signée permettant à un utilisateur de retirer unilatéralement ses fonds depuis Spark (ou une statechain) vers la couche principale, sans nécessiter la coopération de la *Spark Entity*.

L'exit transaction est créée et co-signée par l'utilisateur et la SE au moment où l'utilisateur reçoit des fonds sur Spark. Chaque nouveau propriétaire obtient une exit transaction dotée d'un *timelock* relatif plus court que celui du propriétaire précédent. Cette hiérarchie décroissante de timelocks assure que le propriétaire actuel peut toujours publier sa transaction on-chain avant les anciens propriétaires, empêchant ainsi les tentatives de double dépense. Pour sortir, l'utilisateur diffuse l'exit transaction ainsi que ses transactions parentes sur le réseau Bitcoin.

Termes associés : **STATECHAIN** | **TIMELOCK**

EXPLORATEUR DE BLOC

OUTIL | EN : BLOCK EXPLORER

Outil en ligne ou en local qui permet de transformer les données brutes de la blockchain Bitcoin en un format structuré et facilement lisible par l'Homme. L'explorateur inclut généralement un moteur de recherche afin de localiser facilement un bloc, une transaction ou une adresse spécifique.

Termes associés : **ANALYSE DE CHAINE** | **BLOCKCHAIN**

EXTENSION BLOCK

PROTOCOLE | FR : BLOC D'EXTENSION

Proposition qui vise à augmenter la capacité transactionnelle de Bitcoin par l'ajout d'un bloc secondaire rattaché au bloc principal via un engagement dans la transaction coinbase. Ce concept a été initialement proposé par Johnson Lau en 2013 sous le nom d'« *auxiliary blocks* », puis repris en mars 2017 par Christopher Jeffrey, Joseph Poon, Fedor Indutny et Stephen Pair, en pleine guerre des blocs. Dans ce mécanisme, un mineur inclut dans la coinbase une sortie `OP_RETURN` qui contient la racine de Merkle d'un bloc d'extension. Ce bloc secondaire dispose de son propre ensemble d'UTXO et de ses propres règles de consensus, tout en restant validé par les nœuds complets qui le prennent en charge. Les fonds circulent entre la chaîne principale et le bloc d'extension par des « transactions de résolution » qui consolident les entrées et les sorties entre les deux ensembles.

Cette proposition n'a jamais été formalisée en BIP et n'a pas été activée. Elle a été critiquée pour sa complexité, pour ses effets similaires à ceux d'un hard fork du point de vue économique, et pour l'absence de bénéfices réels par rapport à SegWit.

Termes associés : **BLOCKSIZE WAR** | **SEGWIT** | **SOFT FORK**

EXTRA-NONCE

↗ MINAGE

Champ utilisé dans le `scriptSig` de la transaction coinbase d'un bloc, qui permet d'avoir un plus grand nombre de possibilités à tester pour avoir un hachage inférieur ou égal à la cible de difficulté, en plus du nonce classique qui se trouve, lui, directement dans l'entête de chaque bloc.

Modifier l'extra-nonce dans la transaction coinbase change l'identifiant de cette transaction, et donc la racine de Merkle de toutes les transactions du bloc, ce qui modifie également l'entête du bloc. Cela permet au mineur de continuer à chercher des hachages quand la plage du nonce classique est déjà épuisée, sans pour autant changer la structure de son bloc candidat.

Dans le cadre des pools de minage, l'extra-nonce est souvent divisé en deux parties : une générée par la pool pour identifier chaque hacheur, et une autre modifiée par le hacheur dans la recherche d'une *share* valide. Cela permet aux différents hacheurs de la pool de travailler simultanément sur un même bloc candidat avec l'entièreté de la plage des nonces, sans pour autant dupliquer le même travail au niveau de la pool.

Terme associé : **NONCE**

EXTRA TRANSACTION PROOF

RGB

Dans le cadre du protocole RGB, l'ETP constitue la partie de l'Anchor qui intègre les données complémentaires indispensables à la validation d'un commitment de type Tapret (dans le contexte de Taproot). Elle comprend, entre autres, la clé publique interne associée au script Taproot et les informations spécifiques requises pour le *Script Path Spend*. Cette composante assure ainsi la vérification précise des engagements cryptographiques.

Termes associés : ANCHOR | COMMITMENT

F

F2POOL

MINAGE

Pool de minage de Bitcoin fondée en avril 2013 par Chun Wang et Mao Shixing (« Discus Fish ») en Chine. F2Pool est la toute première pool de minage créée en Chine et l'une des plus anciennes encore en activité au monde. Elle figure régulièrement parmi les pools les plus importantes en termes de hashrate sur le réseau Bitcoin, et s'adresse aussi bien aux mineurs individuels qu'aux infrastructures de minage professionnelles.

Termes associés : **MINAGE** | **MINING POOL**

FARADAY

LIGHTNING NETWORK

Outil développé par Lightning Labs conçu pour extraire des données d'un nœud LND et les analyser afin d'assister son opérateur. Il offre des recommandations pour la fermeture des canaux non performants et fournit des informations détaillées sur le comportement de routage du nœud. Faraday aide à identifier les canaux à faible volume et ceux ayant des problèmes de disponibilité (*uptime*). Cet outil vise à assister les opérateurs de nœuds dans l'allocation de leur capital dans leurs canaux.

Termes associés : **CANAL DE PAIMENT** | **LND**

FAUCET

HISTOIRE | FR : ROBINET

Site web ou application qui distribue gratuitement de petites quantités de bitcoins à ses visiteurs, généralement en échange de la résolution d'un captcha ou d'une tâche simple. Le terme « faucet » (littéralement « robinet ») fait référence à l'idée de distribuer de la monnaie au goutte-à-goutte.

Le premier faucet Bitcoin a été créé par Gavin Andresen en juin 2010 et distribuait 5 BTC par visiteur. Au total, environ 19 700 BTC ont été distribués par ce site, dans le but de faire connaître Bitcoin et d'encourager son adoption à une époque où la cryptomonnaie n'avait quasiment aucune valeur marchande.

Des faucets existent aujourd'hui principalement sur les réseaux de test (testnet, signet), où ils servent à fournir des bitcoins de test aux développeurs. Sur le réseau principal, les faucets ont essentiellement disparu en raison de la valeur élevée du bitcoin et de la prolifération de bots.

Termes associés : **FINNEY HAL** | **NAKAMOTO SATOSHI**

FÉDÉRATION

SIDECHAIN | EN : FEDERATION

Groupe de participants qui collaborent pour administrer un réseau ou un système, en établissant un compromis entre centralisation et décentralisation. Dans une fédération, la responsabilité est répartie entre plusieurs entités indépendantes plutôt que concentrée dans une seule, ce qui élimine les points de défaillance uniques et renforce la résilience du système.

Dans l'écosystème Bitcoin, le modèle fédéré est utilisé notamment par le réseau Liquid, où un ensemble de fonctionnaires signent collectivement les blocs et gèrent les opérations d'ancrage (*peg-in* et *peg-out*) entre la chaîne principale et la sidechain. Le protocole Fedimint repose également sur un modèle fédéré pour la gestion de *Chaumian mints*. Le concept de *Strong Federations*, formalisé par Blockstream, décrit un système dans lequel les fonctionnaires opèrent sous des règles de consensus strictes, sans nécessiter de confiance individuelle envers chaque membre.

Termes associés : **DYNAFED - DYNAMIC FEDERATIONS** | **LIQUID NETWORK**

FEDIMINT

COUCHE SUPÉRIEURE

Protocole de paiement et de gestion de bitcoins conçu pour améliorer la confidentialité et réduire les besoins envers la chaîne principale par la mutualisation de la garde des fonds. Fedimint a été créé par Eric Sirion en 2021. Il s'appuie sur un système de banque chaumienne, qui, au lieu d'être centralisée sur un seul acteur de confiance, s'appuie sur des fédérations. Ces fédérations sont des groupes de gardiens de confiance qui détiennent collectivement et gèrent les bitcoins des utilisateurs de leur groupe. Au sein du groupe, les utilisateurs peuvent réaliser des paiements avec des billets émis en échange de leur dépôt de bitcoins. L'idée de Fedimint est de déployer ce concept au niveau de communautés locales. C'est donc une sorte d'évolution de la banque de dépôt reposant sur le bitcoin, avec le système eCash de David Chaum, et l'utilisation d'une fédération de personnes de confiance en charge du dépôt et de l'émission du sous-jacent.

Termes associés : **ECASH - DAVID CHAUM** | **LIGHTNING NETWORK**

FEE BUMPING

PORTEFEUILLE | FR : AUGMENTATION DES FRAIS

Ensemble de techniques qui permettent d'augmenter les frais d'une transaction déjà diffusée sur le réseau Bitcoin afin d'accélérer sa confirmation. Lorsqu'une transaction reste bloquée dans les mempools en raison de frais trop faibles par rapport à la demande d'espace dans les blocs, les techniques de fee bumping permettent de réévaluer sa priorité auprès des mineurs.

Les deux méthodes existantes sont le RBF (*Replace-By-Fee*) et le CPFP (*Child Pays For Parent*). Avec le RBF, l'expéditeur crée une nouvelle version de la transaction originale avec des frais plus élevés, ce qui remplace la transaction initiale dans les mempools. Avec le CPFP, le destinataire (ou l'expéditeur via un output de change) crée une transaction enfant qui dépense un output de la transaction bloquée, avec des frais suffisants pour inciter les mineurs à confirmer les deux transactions ensemble.

Termes associés : **CPFP - CHILD PAY FOR PARENT** | **RBF - REPLACE-BY-FEE**

FEE_ESTIMATES.DAT

PROTOCOLE

Fichier dans Bitcoin Core qui stocke des données estimées sur les frais de transaction, compilées par le logiciel à partir des transactions récentes et de l'état actuel de la mempool. Ces statistiques aident l'utilisateur à déterminer des frais appropriés à inclure dans ses transactions pour qu'elles soient confirmées en fonction de ses attentes. Ce fichier existe depuis la version 0.10.

Termes associés : **FRAIS DE TRANSACTION** | **MEMPOOL**

FEE RATE - BITCOIN

PROTOCOLE | FR : TAUX DE FRAIS BITCOIN

Mesure du coût unitaire des frais d'une transaction Bitcoin, exprimée en satoshis par octet virtuel (sat/vB). Le *fee rate* détermine la priorité d'une transaction dans le mempool : plus le taux est élevé, plus les mineurs sont incités à l'inclure rapidement dans un bloc. Le montant absolu des frais correspond à la différence entre le total des inputs et le total des outputs d'une transaction. Le *fee rate* est obtenu en divisant ce montant par le poids de la transaction :

$$fee\ rate\ (sats/vB) = \frac{frais\ (sats)}{poids\ (vB)}$$

Ce taux est choisi librement par l'émetteur de la transaction et fonctionne comme un mécanisme d'enchère. En période de congestion, le *fee rate* minimal pour qu'une transaction soit incluse dans un bloc augmente.

Termes associés : **FEE RATE - LIGHTNING** | **FRAIS DE TRANSACTION**

FEE RATE - LIGHTNING

LIGHTNING NETWORK | FR : TAUX DE FRAIS LIGHTNING

Composante proportionnelle des frais de routage sur le Lightning Network. Pour chaque paiement transmis, un nœud peut prélever un *fee rate* calculé en proportion du montant routé, exprimé en parties par million (ppm). Par exemple, un *fee rate* de 100 ppm signifie que le nœud prélève 100 satoshis pour chaque million de satoshis routés. Le *fee rate* complète le *base fee*, qui est un montant fixe par transaction. L'ajustement du *fee rate* est un levier essentiel de la gestion d'un nœud de routage : un taux trop élevé dissuade le trafic, tandis qu'un taux trop bas ne compense pas le coût d'immobilisation de la liquidité dans le canal.

Termes associés : **BASE FEE** | **LIGHTNING NETWORK**

FEE SNIPING

ATTAQUE | FR : CHASSE AUX FRAIS

Scénario d'attaque dans lequel des mineurs cherchent à réécrire un bloc récemment confirmé dans le but de récupérer les frais de transaction qu'il contient, tout en y ajoutant des transactions à frais élevés arrivées entre-temps dans la mempool. L'objectif final de cette attaque pour le mineur est d'augmenter sa rentabilité. Le *fee sniping* peut devenir de plus en plus profitable à mesure que la récompense de bloc diminue et que les frais de transaction représentent une part plus importante dans les revenus des mineurs. Elle peut également être avantageuse lorsque les frais contenus dans le bloc précédent sont nettement supérieurs à ceux du meilleur bloc candidat suivant. Pour simplifier, le mineur est face à ce choix en termes d'incitations :

- Miner de manière normale à la suite du dernier bloc, avec une forte probabilité de remporter une récompense faible ;

*Tenter de miner un bloc antérieur (*fee sniping*), avec une faible probabilité de remporter une récompense élevée.

Cette attaque constitue un risque pour le système Bitcoin, car plus les mineurs l'adoptent, plus les autres mineurs, initialement honnêtes, sont incités à en faire autant. En effet, chaque fois qu'un nouveau mineur s'ajoute à ceux qui tentent un *fee sniping*, la probabilité qu'un des mineurs attaquants réussisse augmente, et la probabilité qu'un des mineurs honnêtes étende la chaîne diminue en contrepartie. Si cette attaque est menée de manière massive et maintenue dans le temps, les confirmations de bloc ne seraient plus un indicateur fiable de l'immuabilité d'une transaction Bitcoin. Cela rendrait potentiellement le système inutilisable.

Pour contrer ce risque, la plupart des logiciels de portefeuille remplissent automatiquement le champ `nLocktime` afin qu'il conditionne la validation de la transaction à l'inclusion dans la prochaine hauteur de bloc. Ainsi, il devient impossible d'inclure la transaction dans une réécriture du bloc précédent. Si l'utilisation massive du `nLocktime` est adoptée par les utilisateurs de Bitcoin, cela réduit considérablement les incitations au *fee sniping*. En effet, cela encourage la progression de la blockchain plutôt que sa réécriture en réduisant les potentiels

bénéfices de celle-ci. Pour les transactions Taproot, le BIP-0326 propose d'utiliser le champ `nSequence` de manière similaire pour obtenir l'effet équivalent à celui du champ `nLocktime` pour les autres types de transactions. Cette utilisation permettrait de faire d'une pierre deux coups en améliorant également la confidentialité des protocoles de seconde couche qui utilisent ce même champ.

Termes associés : **FRAIS DE TRANSACTION** | **NLOCKTIME**

FEELER CONNECTION

🌐 RÉSEAU | FR : CONNEXION SONDE

Connexion sortante éphémère que Bitcoin Core établit toutes les deux minutes, une fois les 10 connexions sortantes permanentes remplies (8 classiques et 2 block-relay-only). Elle se limite au handshake `VERSION`/`VERACK` pour vérifier qu'une adresse correspond bien à un nœud Bitcoin joignable, puis se ferme.

La feeler connection alterne entre deux comportements. Si le tampon d'éviction contient des adresses (cas d'une collision lors de l'insertion dans la table « tried »), elle en teste une : si l'adresse existante répond, la nouvelle est ignorée ; sinon, l'ancienne est remplacée. Ce mécanisme est appelé *test-before-evict*. Si le tampon est vide, la feeler tire une adresse aléatoire de la table « new » et tente de s'y connecter : en cas de succès, l'adresse est promue dans « tried ».

Ce mécanisme a été introduit par Ethan Heilman comme contre-mesure à certains types d'attaque éclipse. En peuplant la table « tried » avec des adresses récemment vérifiées, il réduit la proportion d'adresses obsolètes et rend plus difficile l'éclipse d'un nœud. Lorsqu'aucun nouveau bloc n'a été reçu depuis 30 minutes, Bitcoin Core détecte un éventuel *stale tip* et établit une connexion sortante supplémentaire vers un nouveau pair, afin de vérifier s'il dispose d'un bloc plus récent et de contourner une éventuelle partition du réseau. Ce mécanisme est distinct des feeler connections.

Termes associés : **ECLIPSE** | **PAIR SORTANT**

FERME DE MINAGE

↗ MINAGE | EN : MINING FARM

Installation où de nombreuses machines de minage (souvent, des ASICs) sont regroupées pour miner du bitcoin en participant au processus de la preuve de travail. Le but de ce regroupement est de faciliter la gestion du parc de machines et de faire des économies d'échelles, notamment pour la mise en place, l'entretien, le refroidissement, la fourniture en électricité et la connexion au réseau. La ferme de minage permet également d'améliorer la proximité des mineurs, et donc de réduire la latence qui pourrait affecter leur rentabilité.

Attention, la ferme de minage ne doit pas être confondue avec la pool de minage.

Termes associés : **ASIC** | **MINING POOL**

FIAT

ÉCONOMIE ET RÉGULATION

Monnaie, souvent étatique, dont le cours est imposé par la force publique.

Le terme de « fiat » est parfois traduit par « fiduciaire » bien que ce dernier terme ne prenne pas en compte la dimension de violence légitime qu'incarne le terme « fiat ». En français, il est souvent admis d'utiliser directement le terme anglais de « fiat ».

Termes associés : **CRYPTOMONNAIE** **INFLATION**

FIBRE

RÉSEAU

Sigle de « *Fast Internet Bitcoin Relay Engine* ». C'est un protocole conçu par Matt Corallo en 2016 pour accélérer la diffusion des blocs Bitcoin à travers le monde. Son objectif était de réduire les délais de propagation au plus près des limites physiques. FIBRE visait à garantir une distribution plus équitable des opportunités de minage, en s'assurant que la proportion de blocs minés par un participant reflète fidèlement sa contribution en termes de puissance de calcul, peu importe sa situation sur le réseau.

En effet, la latence dans la transmission des blocs peut favoriser les grands groupes de mineurs, bien connectés et souvent à proximité, au détriment des plus modestes. Ce phénomène pourrait, à terme, augmenter la centralisation du minage et réduire la sécurité globale du système. Pour pallier ce problème, FIBRE introduisait des codes de correction d'erreur et l'envoi de données supplémentaires pour contrebalancer les pertes de paquets, ainsi que l'utilisation de blocs compactés similaires à ceux décrits dans le BIP-0152, le tout opérant via UDP pour contourner certaines limitations de TCP. Néanmoins, FIBRE fut délaissé en 2020, principalement en raison de sa dépendance à l'égard d'un unique mainteneur et du fait que l'adoption du BIP-0152 a rendu un tel système moins indispensable.

Terme associé : **BIP-0152**

FIDELITY BOND

CONFIDENTIALITÉ

Mécanisme par lequel un utilisateur sacrifie délibérément de la valeur en bitcoins pour rendre une identité cryptographique coûteuse à obtenir, ce qui dissuade les attaques Sybil. Le concept, proposé initialement par Peter Todd en 2013, a trouvé son application principale dans JoinMarket, où il renforce la confidentialité des coinjoins.

Dans JoinMarket, un *maker* qui publie un *fidelity bond* de valeur élevée a une probabilité plus grande d'être sélectionné par les *takers* pour participer aux coinjoins, et donc de percevoir des frais. Le moyen le plus courant de créer un *fidelity bond* est d'envoyer des bitcoins vers une adresse verrouillée dans le temps via un `OP_CHECKLOCKTIMEVERIFY`. Ce qui est sacrifié n'est pas le capital lui-même, mais sa valeur temporelle : les bitcoins restent la propriété du *maker*, mais sont immobilisés jusqu'à l'expiration du verrou. Un détenteur long terme peut ainsi

constituer un *fidelity bond* à coût quasi nul, puisqu'il ne comptait pas dépenser ses bitcoins avant l'échéance.

Une alternative consiste à détruire des bitcoins en les envoyant vers une sortie `OP_RETURN`, rendant le sacrifice permanent et la preuve triviale à vérifier.

Termes associés : **COINJOIN** | **JOINMARKET**

FINALITÉ DE TRANSACTION

PROTOCOLE | EN : TRANSACTION FINALITY

Concept qui désigne le moment à partir duquel une transaction est considérée comme irréversible et définitivement inscrite dans la blockchain. Sur Bitcoin, la finalité est probabiliste plutôt qu'absolue : chaque nouveau bloc ajouté au-dessus du bloc qui contient la transaction rend sa réversion exponentiellement plus difficile et coûteuse. Par convention, une transaction est généralement considérée comme finale après 6 confirmations, soit environ une heure, car la probabilité qu'un attaquant puisse réorganiser la chaîne, ou bien qu'un embranchement naturel arrive sur une telle profondeur est extrêmement faible. Cependant, pour des montants modestes, une ou deux confirmations peuvent être jugées suffisantes.

Termes associés : **CONFIRMATION** | **DOUBLE DÉPENSE**

FINNEY HAL

HISTOIRE

Harold T. Finney II, dit Hal Finney, est un cryptographe et développeur célèbre pour son rôle crucial dans les débuts de Bitcoin et ses contributions à la cryptographie. Dès la publication du White Paper de Bitcoin en 2008, il fut l'un des premiers à interagir avec Satoshi Nakamoto. Il a apporté des retours, signalé des bugs et proposé des améliorations après le lancement du logiciel en janvier 2009. Il a été le destinataire de la première transaction Bitcoin (en dehors des coinbases), en recevant 10 BTC de la part de Satoshi dans le bloc n° 170 :

```
f4184fc596403b9d638783cf57adfe4c75c605f6356fbc91338530e9831e9e16
```

Hal Finney est aussi probablement la première personne, après Satoshi, à avoir miné un bloc : le bloc n° 78. Plus que cela, Hal Finney a été le premier promoteur de Bitcoin durant une période où le projet était encore méconnu.

En dehors de Bitcoin, il est également reconnu pour son invention de RPoW (*Reusable Proofs of Work*), un système de monnaie électronique lancé en 2004. Bien que RPoW n'ait pas rencontré le succès attendu, il demeure l'un des précurseurs les plus aboutis de Bitcoin. En tant que cypherpunk engagé, Hal Finney a également joué un rôle déterminant dans l'élaboration et l'amélioration de PGP (*Pretty Good Privacy*). Hal Finney nous a quittés le 28 août 2014, emporté par la sclérose latérale amyotrophique (maladie de Charcot). Il a été cryogénisé par la fondation Alcor. Il restera une figure majeure de l'histoire de la cryptographie et de Bitcoin.

Termes associés : **RPOW** | **WHITE PAPER**

FLAG DAY

PROTOCOLE

Méthode d'activation de soft fork utilisée dans les premières années de Bitcoin. Ce processus définit simplement une date butoir, connue sous le nom de « *Flag Day* », avant laquelle la mise à jour du protocole doit être adoptée par l'ensemble du réseau. Cette approche est simple et directe : après cette date, les nœuds et les mineurs doivent avoir mis à jour leur logiciel pour se conformer aux nouvelles règles, sans quoi ils risquent de se retrouver sur une chaîne incompatible.

Cependant, cette méthode est très risquée de nos jours, car elle nécessite une coordination et un consensus important au sein de la communauté, faute de quoi le réseau peut subir une scission, et la chaîne à jour peut ne pas être la plus longue (avec le plus de travail accumulé). La méthode du *Flag Day* peut toutefois être utilisée pour des changements non controversés ou des rectifications techniques urgentes.

Termes associés : **SOFT FORK** | **UASF**

FLOOD AND LOOT

ATTAQUE

Attaque sur le Lightning Network qui exploite le mécanisme des HTLC pour voler des fonds. Elle a été décrite en 2020 par Jona Harris et Aviv Zohar. L'attaquant contrôle deux nœuds et ouvre des canaux avec de nombreuses victimes intermédiaires. Il route un grand nombre de HTLC depuis son nœud source vers son nœud cible en passant par les victimes, puis refuse de révéler la préimage. À l'approche de l'expiration des *timelocks*, toutes les victimes sont contraintes de fermer unilatéralement leurs canaux au même moment, ce qui provoque une congestion massive de la blockchain. Cette congestion empêche certaines transactions de fermeture d'être confirmées à temps. Les HTLC non réclamés avant l'expiration reviennent à l'attaquant, qui peut surenchérir sur les frais grâce à RBF. Contrairement au *griefing*, qui gèle temporairement des fonds sans les voler, le *flood-and-loot* permet un vol effectif. Les chercheurs ont estimé qu'environ 85 canaux attaqués simultanément suffisent pour garantir un gain. L'attaque n'a jamais été observée en conditions réelles. Les contre-mesures proposées incluent la réduction du nombre maximal de HTLC par canal, l'adoption des *anchor outputs* pour ajuster dynamiquement les frais lors d'une fermeture forcée, et le raccourcissement des délais d'expiration.

Termes associés : **FORCE CLOSE** | **GRIEFING ATTACK** | **HTLC** | **LIGHTNING NETWORK**

FONCTION DE HACHAGE

CRYPTOGRAPHIE | EN : HASH FUNCTION

Fonction mathématique qui prend une entrée de taille variable (appelée message) et produit une sortie de taille fixe (appelée hash, hachage, condensat ou empreinte). Les fonctions de hachage sont des primitives largement utilisées en cryptographie. Elles présentent des propriétés spécifiques qui les rendent appropriées pour une utilisation dans des contextes sécurisés :

- Résistance aux préimages : il doit être très difficile de trouver un message

donnant un hachage spécifique, c'est-à-dire de trouver une préimage m pour un hash h tel que $h = H(m)$, où H est la fonction de hachage ;
- Résistance aux secondes préimages : étant donné un message m_1, il doit être très difficile de trouver un autre message m_2 (différent de m_1) tel que $H(m_1) = H(m_2)$;
- Résistance aux collisions : il doit être très difficile de trouver deux messages distincts m_1 et m_2 tels que $H(m_1) = H(m_2)$;
- Résistance à la falsification : de petites modifications dans l'entrée doivent provoquer des changements significatifs et imprévisibles dans la sortie.

Dans le contexte de Bitcoin, les fonctions de hachage sont utilisées à plusieurs fins, notamment pour le mécanisme de preuve de travail (*Proof-of-Work*), les identifiants de transaction, la génération d'adresses, le calcul de sommes de contrôle et la création de structures de données telles que les arbres de Merkle. Sur la partie protocolaire, Bitcoin utilise exclusivement la fonction `SHA256d`, également nommée `HASH256`, qui consiste en un double hachage `SHA256`. On utilise aussi `HASH256` dans le calcul de certaines sommes de contrôle, notamment pour les clés étendues (`xpub`, `xprv`...). Sur la partie portefeuille, on utilise également :

- `SHA256` simple pour les sommes de contrôle des phrases mnémoniques ;
- `SHA512` au sein des algorithmes `HMAC` et `PBKDF2` utilisés dans le processus de dérivation des portefeuilles déterministes et hiérarchiques ;
- `HASH160`, qui décrit une utilisation successive d'un `SHA256` et d'un `RIPEMD160`. `HASH160` est utilisé dans le processus de génération des adresses de réception (sauf P2PK, P2WSH et P2TR) et dans le calcul des empreintes de clés parents pour les clés étendues.

Termes associés : **ARBRE DE MERKLE** | **SHA256**

FONCTION DE HACHAGE TAGUÉE

CRYPTOGRAPHIE | EN : TAGGED HASH FUNCTION

Variante de fonction de hachage utilisée dans le protocole Bitcoin (notamment dans Taproot) pour garantir la séparation des domaines. Une fonction de hachage taguée prend un tag (une chaîne de caractères UTF-8) et des données en entrée, et calcule le hachage comme suit :

$$TaggedHash(tag,\ data) = SHA256(SHA256(tag) \parallel SHA256(tag) \parallel data)$$

Le fait de préfixer les données par le double hachage du tag garantit que des hachages calculés dans des contextes différents ne peuvent pas entrer en collision. Par exemple, un hachage calculé avec le tag `TapLeaf` ne pourra jamais être identique à un hachage calculé avec le tag `TapBranch`, même si les données sont les mêmes.

Ce mécanisme est introduit par le BIP-0340 et utilisé dans l'ensemble des BIPs liés à Taproot (BIP-0341 et BIP-0342). Chaque étape de la construction et de la dépense d'une sortie P2TR utilise un tag distinct : `TapLeaf` pour les feuilles de l'arbre de scripts, `TapBranch` pour les branches, `TapTweak` pour le calcul

du tweak, et `TapSighash` pour la création du hash de signature. Le BIP-0340 utilise aussi des tags pour les signatures de Schnorr : `BIP0340/challenge`, `BIP0340/aux` et `BIP0340/nonce`.

Termes associés : **FONCTION DE HACHAGE** | **TWEAK**

FONCTIONNAIRE

SIDECHAIN | EN : FUNCTIONARIES

Dans le cadre de la sidechain Liquid, les fonctionnaires sont des nœuds pilotés par des entités chargées de gérer le système. Ils ont principalement deux rôles : établir le consensus et exécuter des transactions en tant que signataire de bloc (*blocksigners*), ainsi que sécuriser les bitcoins détenus par le réseau afin d'assurer l'ancrage bilatéral (*watchmen*).

Termes associés : **BLOCKSIGNERS** | **LIQUID NETWORK**

FONDATION BITCOIN

ORGANISATION | EN : BITCOIN FOUNDATION

Organisation à but non lucratif fondée en septembre 2012. Ses membres fondateurs sont Gavin Andresen, Charlie Shrem, Mark Karpelès, Peter Vessenes, Roger Ver et Patrick Murck. Elle avait pour objectif de promouvoir le développement de Bitcoin, de financer les développeurs du protocole et de représenter la communauté auprès des institutions.

Dans ses premières années, la fondation a financé le travail de développeurs comme Gavin Andresen et contribué à la standardisation du protocole. Toutefois, elle a rapidement été confrontée à des controverses. Plusieurs de ses membres fondateurs ont fait l'objet de poursuites judiciaires : Charlie Shrem a été arrêté en 2014 pour blanchiment de capitaux lié au marché Silk Road, et Mark Karpelès a été inculpé au Japon après la faillite de Mt. Gox.

La fondation a perdu de son influence au fil des années, critiquée pour sa gouvernance et son manque de transparence financière. Une partie de la communauté Bitcoin lui reprochait de vouloir centraliser la représentation d'un projet conçu pour être décentralisé.

Termes associés : **BITCOIN CORE** | **BLOCKSIZE WAR**

FONGIBILITÉ

ÉCONOMIE ET RÉGULATION | EN : FUNGIBILITY

Propriété d'une monnaie qui assure que chaque unité est interchangeable et indistinguable d'une autre unité similaire. Le bitcoin, en tant qu'unité de compte, est en principe fongible, car un bitcoin vaut toujours un autre bitcoin : 1 BTC = 1 BTC. Cependant, la traçabilité des UTXOs (le support des unités) sur la blockchain peut parfois compromettre cette fongibilité. En effet, chaque satoshi peut être distingué par son historique, ce qui lui confère ainsi une identité spécifique. La perception de l'historique de chaque UTXO peut influencer le jugement des parties sur la fongibilité des bitcoins utilisés. Ainsi, bien que la fongibilité soit une caractéristique intrinsèque des unités monétaires, elle peut être altérée par les spécificités du support utilisé pour ces unités, comme c'est

le cas avec Bitcoin.

Termes associés : **COINJOIN** | **UTXO**

FORCE BRUTE

ATTAQUE | EN : BRUTE FORCE ATTACK

Méthode de cryptanalyse pour trouver un mot de passe ou une clé qui consiste à essayer par tâtonnement toutes les combinaisons possibles de clés ou de mots de passe jusqu'à trouver celle qui permet d'accéder à un privilège ou une information protégée. Cette technique repose sur du calcul intensif et peut être extrêmement longue, surtout face à des clés de grande taille. Pour faire face à ce type d'attaque, il faut utiliser des séquences de mot de passe et de clés plus longues afin de multiplier le nombre d'opérations nécessaires pour l'attaquant. En théorie, la complexité d'une telle attaque est exponentielle en la longueur de la cible.

Termes associés : **CRYPTANALYSE** | **ENTROPIE**

FORCE CLOSE

LIGHTNING NETWORK | FR : FERMETURE FORCÉE

Mécanisme de fermeture non coopérative d'un canal Lightning. Lorsque deux utilisateurs ouvrent un canal avec un multisig 2/2, chacun peut unilatéralement fermer le canal en diffusant la dernière transaction d'engagement qui est déjà signée, afin de récupérer ses bitcoins *onchain*. Dans ce cas, on parle d'un « *force close* ».

Cette méthode est généralement utilisée si l'un des participants ne répond plus ou si la fermeture coopérative du canal est impossible. Si l'on peut éviter le *force close*, c'est mieux, car il nécessite plus de temps pour récupérer ses fonds *onchain* et peut couter beaucoup plus cher en frais.

Lors d'un *force close*, un des deux utilisateurs diffuse la transaction d'engagement qui reflète le dernier état connu du canal Lightning. Il y a ensuite un délai de sécurité (*timelock*) avant que les bitcoins puissent être récupérés *onchain*, ce qui laisse le temps à l'autre partie de vérifier que la transaction correspond bien au dernier état du canal. Si un utilisateur tente de tricher en publiant une transaction d'engagement désuète, l'autre partie peut utiliser le secret de révocation pour punir le tricheur et récupérer la totalité des fonds du canal.

Termes associés : **TIMELOCK** | **TRANSACTION D'ENGAGEMENT**

FORCED ADDRESS REUSE

ATTAQUE | FR : RÉUTILISATION D'ADRESSE FORCÉE

Attaque qui consiste à envoyer de minuscules quantités de bitcoins à un grand nombre d'adresses de réception. L'objectif de l'attaquant est de pousser les destinataires à regrouper ces sommes avec d'autres UTXOs. L'attaquant suit ensuite les déplacements futurs de ces faibles quantités de bitcoins, dans le but de former des *clusters* d'adresses, c'est-à-dire de déterminer si plusieurs adresses appartiennent à une même entité. En croisant les informations recueillies lors de l'attaque avec d'autres données et heuristiques utilisées dans l'analyse de chaîne, il est possible pour l'attaquant d'identifier certaines entités et les adresses associées. Cette méthode représente une menace uniquement pour la confidentialité des utilisateurs, mais n'affecte pas la sécurité de leurs fonds.

Le terme original pour décrire cette attaque est « Dusting Attack », mais certains bitcoiners suggèrent plutôt d'utiliser le terme de « forced address reuse », car ils trouvent que le terme de « dust » est ici inapproprié.

Termes associés : **DUST** | **DUST LIMIT**

FORFEIT TRANSACTION

COUCHE SUPÉRIEURE | FR : TRANSACTION DE FORFAIT

Transaction par laquelle un utilisateur du protocole Ark cède la propriété de son VTXO d'entrée à l'*Ark Service Provider* (ASP) lors d'un round. L'ASP avançant la liquidité on-chain pour créer les nouveaux VTXO de sortie, il récupère en contrepartie les anciens VTXO des utilisateurs via ces transactions de forfait.

La forfeit transaction exploite la clause de forfait (*forfeit clause*) inscrite dans la politique du VTXO, qui consiste en un multisig entre l'utilisateur et l'ASP. Sans mécanisme de protection supplémentaire, signer cette transaction reviendrait à céder son VTXO sans aucune garantie d'obtenir un nouveau VTXO en retour. Pour résoudre ce problème, un *connector output* issu de la transaction de round est ajouté comme entrée supplémentaire dans la forfeit transaction. Ce lien rend la transaction invalide tant que la transaction de round n'est pas confirmée on-chain, garantissant ainsi l'atomicité : l'utilisateur ne perd son ancien VTXO que si les nouveaux VTXO du round sont effectivement créés.

Le mécanisme de la forfeit transaction est également repris dans le cadre des paiements OOR (*out-of-round*), où la clause de forfait est réutilisée pour transférer un VTXO directement à un autre utilisateur sans passer par un round.

Termes associés : **CONNECTOR** | **ROUND ARK**

FORK

PROTOCOLE

Le terme de « fork » revêt plusieurs significations dans le cadre de Bitcoin. Il désigne soit une scission du réseau de nœuds en plusieurs groupes séparés, entraînant la création de plusieurs blockchains différentes, soit une modification des règles du protocole, voire les deux simultanément. Pour simplifier, on distingue 4 grandes catégories de forks :

- L'embranchement naturel : se produit lorsqu'il y a une concurrence temporaire entre deux blocs découverts en même temps à une même hauteur. Cet embranchement peut s'étendre sur plusieurs blocs. Ce type de fork se résout naturellement quand une des chaînes devient plus longue que l'autre (avec plus de travail accumulé), menant à une réorganisation. Cette réorganisation se manifeste avec l'intégralité des nœuds qui s'accordent de nouveau sur une blockchain unique ;
- Le fork de code : consiste à créer une toute nouvelle cryptomonnaie à partir du code source de Bitcoin, en lançant une nouvelle blockchain depuis le bloc de Genèse ;
- Le hard fork : représente une modification du protocole Bitcoin, incompatible avec les versions antérieures, en retirant des règles ou en allégeant celles existantes. Il en résulte la création de deux chaînes distinctes et incompatibles si tous les nœuds ne sont pas mis à jour. Le réseau se scinde alors en deux : ceux qui adoptent les nouvelles règles et ceux qui conservent les anciennes ;
- Le soft fork : implique des modifications rétrocompatibles qui ajoutent des règles ou rendent plus restrictives celles existantes, sans provoquer de division du réseau. Les nœuds qui n'adoptent pas les nouvelles règles peuvent continuer à suivre la même chaîne que les autres, à condition que la majorité de la puissance de calcul du réseau soutienne la chaîne mise à jour.

Termes associés : **HARD FORK** | **SOFT FORK**

FORK - GIT

</> INFORMATIQUE

Dans le cadre de Git, représente la création d'une copie d'un dépôt existant sur un nouveau compte, permettant ainsi à l'utilisateur de modifier, tester ou développer le projet indépendamment du dépôt original. Les forks permettent la collaboration open source et la contribution à des projets sans affecter le dépôt source.

Termes associés : **GIT** | **GITHUB**

FOSS

‹/› INFORMATIQUE

Acronyme de « *Free and Open Source Software* ». Cela désigne un logiciel sous licence autorisant quiconque à l'exécuter pour tout usage, à l'étudier, à le modifier pour l'adapter à ses besoins, et à le redistribuer librement, modifié ou non. Cette définition implique que le code source du logiciel soit accessible publiquement.

Termes associés : **BITCOIN CORE** | **GIT**

FOUNDATION DEVICES

ORGANISATION

Entreprise américaine cofondée en 2020 par Zach Herbert, Jacob Johnston, Ken Carpenter et Matt Beach à Boston, spécialisée dans la fabrication de matériel open source pour la sécurisation de bitcoins. Son produit phare, le Passport, est un portefeuille matériel (*hardware wallet*) fonctionnant de manière air-gapped via des QR codes et des cartes microSD, sans aucune connexion réseau directe. Foundation Devices développe également Envoy, une application mobile compagnon qui sert d'interface pour gérer le Passport et interagir avec le réseau Bitcoin. L'ensemble du matériel est assemblé aux États-Unis et le firmware est entièrement open source.

Terme associé : **HARDWARE WALLET**

FOUNDRY USA

↗ MINAGE

Pool de minage de Bitcoin basée aux États-Unis, exploitée par Foundry Digital, une filiale de Digital Currency Group (DCG). Lancée en 2020, Foundry USA Pool est devenue l'une des plus grandes pools de minage au monde en termes de hashrate, occupant régulièrement la première place. Elle fournit des services de minage aux opérateurs institutionnels et aux grandes fermes de minage nord-américaines. Foundry propose également des services de financement, d'achat de matériel et de conseil pour les mineurs professionnels. La prédominance de Foundry USA dans la distribution du hashrate de Bitcoin soulève des questions sur la centralisation géographique du minage, la majorité de sa puissance de calcul étant concentrée aux États-Unis.

Termes associés : **HASHRATE** | **MINING POOL**

FOUNTAIN

✱ OUTIL

Application de podcast intégrant des paiements Lightning selon le modèle *value-for-value*. Fondée en 2021 par Oscar Merry et Nick Malster, Fountain permet aux auditeurs d'envoyer des micropaiements en satoshis directement aux créateurs de contenu en temps réel pendant l'écoute, via le Lightning Network. L'application repose sur le standard Podcasting 2.0 initié par Adam Curry, qui intègre nativement les paiements Bitcoin dans les flux RSS. Les auditeurs peuvent également gagner des satoshis en écoutant des podcasts, en recommandant du contenu ou en interagissant avec la communauté. Fountain est disponible sur iOS et Android.

Terme associé : **LIGHTNING NETWORK**

FPGA

↗ MINAGE | FR : RÉSEAU DE PORTES PROGRAMMABLES

Sigle de « *Field-Programmable Gate Array* ». Circuit intégré qui peut être reconfiguré après sa fabrication pour exécuter des opérations spécifiques. Contrairement aux processeurs classiques (CPU), conçus pour être polyvalents, les FPGA permettent de programmer directement la logique matérielle pour l'adapter à une tâche précise, ce qui les rend plus efficaces en termes de consommation énergétique et de performances pour cette tâche.

Dans l'histoire du minage de Bitcoin, les FPGA ont représenté une étape intermédiaire entre le minage par GPU et l'arrivée des ASIC. À partir de 2011, des mineurs ont commencé à utiliser des FPGA pour calculer des hachages `SHA256`, car ils offraient un meilleur rapport performance/consommation électrique que les GPU, tout en restant reprogrammables. L'adoption des FPGA a toutefois été relativement courte, car les ASIC, des circuits entièrement dédiés au hachage `SHA256`, les ont rapidement surpassés en termes de puissance de calcul et d'efficacité énergétique.

Termes associés : **ASIC** | **MINAGE**

FPPS - FULL PAY PER SHARE

↗ MINAGE

Méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. C'est une évolution de la méthode PPS (*Pay Per Share*). Elle rémunère les mineurs non seulement pour chaque share valide qu'ils soumettent, mais inclut également une part des frais de transaction. La rémunération est calculée sur la base des frais de transaction moyens précédents et du hashrate de la pool. Ainsi, les mineurs reçoivent une rétribution pour les shares soumises, qu'un bloc soit trouvé ou non, mais cette méthode rémunère aussi la valeur attendue. Elle offre une rémunération stable et prévisible pour les mineurs, car elle élimine la variance liée à la probabilité de trouver un bloc, tout en les exposant aux fluctuations du marché de frais. Toutefois, elle est plus risquée pour les opérateurs de pool, car ils doivent payer les mineurs même lorsque aucun bloc n'est trouvé, absorbant ainsi le risque de variance.

Terme associé : **SHARES**

FRAIS DE TRANSACTION

PROTOCOLE | EN : TRANSACTION FEES

Les frais de transaction représentent une somme qui vise à rémunérer les mineurs pour leur participation au mécanisme de la preuve de travail. Ces frais incitent les mineurs à inclure les transactions dans les blocs qu'ils créent. Ils sont le résultat de la différence entre le montant total des inputs et le montant total des outputs d'une transaction :

```
frais = inputs - outputs
```

Ils sont exprimés en `sats/vBytes`, ce qui veut dire que les frais ne dépendent pas du montant des bitcoins envoyés, mais du poids de la transaction. Ils sont choisis librement par l'émetteur d'une transaction et déterminent sa vitesse d'inclusion dans un bloc par un mécanisme d'enchère. Par exemple, imaginons que je réalise une transaction avec un input de `100 000 sats`, un output de `40 000 sats` et un output de `58 500 sats`. Le total des outputs est de `98 500 sats`. Les frais alloués à cette transaction sont de `1 500 sats`. Le mineur qui inclut ma transaction pourra créer `1 500 sats` dans sa transaction coinbase en contrepartie des `1 500 sats` que je n'ai pas récupérés dans mes outputs.

Les transactions avec des frais plus élevés, en fonction de leur taille, sont traitées en priorité par les mineurs, ce qui peut accélérer le processus de confirmation. Inversement, les transactions avec des frais plus faibles peuvent être retardées lors des périodes de forte congestion. Il convient de noter que les frais de transaction Bitcoin sont distincts de la subvention de bloc, qui est une incitation supplémentaire pour les mineurs. La récompense de bloc est composée de nouveaux bitcoins créés à chaque bloc miné (subvention de bloc), ainsi que des frais de transaction collectés. Tandis que la subvention de bloc diminue au fil du temps en raison de la limitation de l'offre totale de bitcoins, les frais de transaction, eux, continueront de jouer un rôle crucial pour encourager les mineurs à participer.

Au niveau protocolaire, rien n'empêche les utilisateurs d'inclure des transactions sans aucuns frais dans un bloc. En réalité, ce type de transaction sans frais fait exception. Par défaut, les nœuds Bitcoin ne relaient pas les transactions disposant de frais inférieurs à `1 sat/vBytes`. Si certaines transactions sans frais ont pu passer, c'est parce qu'elles ont été intégrées directement par le mineur gagnant, sans parcourir le réseau de nœuds. Par exemple, la transaction suivante n'inclut aucuns frais :

```
fd456524104a6674693c29946543f8a0befccce5a352bda55ec8559fc630f5f3
```

Dans cet exemple précis, c'était une transaction initiée par le directeur de la pool de minage F2Pool. En tant qu'utilisateur normal, la limite inférieure est donc actuellement de `1 sat/vBytes`.

Il convient également de tenir compte des limites de purge. En période de forte congestion, les mempools des nœuds purgent leurs transactions en attente en dessous d'un certain seuil, afin de respecter leur limite de RAM attribuée. Cette limite est librement choisie par l'utilisateur, mais beaucoup laissent la valeur

de Bitcoin Core par défaut à 300 Mo. Elle peut être modifiée dans le fichier `bitcoin.conf` avec le paramètre `maxmempool`.

Termes associés : **MEMPOOL** | **RÉCOMPENSE DE BLOC**

FROST

CRYPTOGRAPHIE | FR : SIGNATURES SCHNORR A SEUIL

Protocole de signatures à seuil Schnorr proposé par Chelsea Komlo et Ian Goldberg en 2020, dont l'acronyme signifie *Flexible Round-Optimized Schnorr Threshold Signatures*. FROST optimise le processus de signature collaborative en le réduisant à deux tours de communication entre les participants, voire à un seul tour lorsqu'un prétraitement des nonces est effectué.

Le protocole repose sur un schéma à seuil (t, n) : parmi n participants qui détiennent chacun un fragment de la clé privée, au moins t d'entre eux doivent coopérer pour produire une signature valide. La signature produite est une signature Schnorr standard conforme au BIP-0340, indiscernable d'une signature émise par un paiement classique. Cette propriété est particulièrement intéressante dans le contexte de Taproot, car elle permet de réaliser des opérations multisignatures on-chain sans révéler la structure à seuil sous-jacente, ce qui améliore la confidentialité et réduit les frais de transaction.

Termes associés : **FROSTSNAP** | **SCHNORR**

FROSTSNAP

OUTIL

Frostsnap est un système de sécurité pour bitcoins reposant sur le protocole FROST (*Flexible Round-Optimized Schnorr Threshold Signatures*). Contrairement au multisig classique par script Bitcoin, FROST produit une unique signature Schnorr *on-chain*, indistinguable d'une transaction à signature simple, ce qui améliore la confidentialité et réduit les frais de transaction. L'équipe de Frostsnap, composée notamment de Nick Farrow et Lloyd Fournier, développe des dispositifs matériels dédiés qui se connectent en chaîne via USB-C. Le système permet d'ajouter ou de retirer des signataires après la génération des clés, sans modifier l'adresse publique. Le matériel et le logiciel sont entièrement *open source* sous licence MIT.

Termes associés : **MULTISIG** | **SCHNORR**

FULCRUM

OUTIL

Implémentation d'un serveur Electrum développée en C++ par Calin Culianu, initialement pour Bitcoin Cash, et publiée sous licence GPLv3. Fulcrum indexe l'intégralité des transactions de la blockchain Bitcoin à partir d'un nœud complet et répond aux requêtes des portefeuilles compatibles avec le protocole Electrum.

Conçu comme un remplacement performant d'ElectrumX, Fulcrum utilise une base de données RocksDB et tire parti de la programmation multithreadée et asynchrone de C++20 pour offrir des temps de réponse rapides, même sous forte charge. Il nécessite un nœud Bitcoin Core (ou Bitcoin Knots) configuré avec `txindex=1` et sans élagage.

La synchronisation initiale de l'index avec la blockchain est bien plus longue qu'avec Electrs, et son stockage avoisine actuellement les 200 Go. En contrepartie, il est nettement plus rapide à l'usage, là où Electrs est plus lent, mais se distingue par sa légèreté.

Termes associés : **ELECTRS** | **ELECTRUM SERVER**

FULL RBF

PROTOCOLE

Politique de mempool qui permet le remplacement de toute transaction non confirmée par une autre transaction qui dépense au moins un même input, mais qui offre des frais plus élevés, indépendamment du fait que la transaction originale ait signalé ou non sa disponibilité au remplacement via le *flag* RBF (champ `nSequence`). Dans le RBF classique (opt-in RBF), seules les transactions qui ont explicitement activé ce signal peuvent être remplacées. Le Full RBF supprime cette restriction.

Cette politique a été introduite dans Bitcoin Core via l'option de configuration `mempoolfullrbf`, disponible à partir de la version 24.0 (novembre 2022) avec une valeur par défaut à `0` (désactivée). À partir de la version 28.0 (octobre 2024), cette option a été activée par défaut (`mempoolfullrbf=1`), ce qui fait du Full RBF le comportement standard de Bitcoin Core.

Le Full RBF a fait l'objet de débats intenses dans la communauté. Ses partisans argumentent qu'il améliore l'efficacité du marché des frais et renforce la sécurité en éliminant la fausse impression que les transactions non signalées RBF seraient irrévocables avant confirmation. Ses opposants soulignaient qu'il complique les cas d'usage qui reposaient sur l'acceptation de transactions non confirmées (transactions « zero-conf »), notamment dans le commerce de détail.

Termes associés : **MEMPOOL** | **RBF - REPLACE-BY-FEE**

G

GALOY

ORGANISATION

Entreprise fondée en 2019 par Nicolas Burtey et Chris Hunter, développant une infrastructure bancaire Bitcoin open source. Galoy est à l'origine du portefeuille Blink (anciennement « Bitcoin Beach Wallet »), l'application qui a servi de socle à l'économie circulaire Bitcoin d'El Zonte au Salvador, avant que le pays n'adopte le bitcoin comme monnaie légale en 2021. La plateforme Galoy permet à des organisations de déployer leur propre infrastructure bancaire Bitcoin et Lightning, soit en auto-hébergement, soit en tant que service géré (*Banking-as-a-Service*). L'entreprise développe également des solutions de prêts adossés à du bitcoin.

Terme associé : **LIGHTNING NETWORK**

GAP LIMIT

PORTEFEUILLE | FR : LIMITE DE CREUX

Paramètre utilisé dans les logiciels de portefeuille Bitcoin pour déterminer le nombre maximal d'adresses consécutives non utilisées à générer avant de cesser la recherche de transactions supplémentaires. L'ajustement de ce paramètre est souvent nécessaire lors de la récupération d'un portefeuille pour garantir que toutes les transactions soient bien trouvées. Un Gap Limit insuffisant pourrait entraîner l'oubli de certaines transactions si des adresses étaient ignorées lors des phases de dérivation. Augmenter le Gap Limit permet au portefeuille de rechercher plus loin dans la séquence d'adresses, afin de récupérer toutes les transactions associées.

En effet, une seule `xpub` peut théoriquement dériver plus de 4 milliards d'adresses de réception (adresses internes et externes). Toutefois, les logiciels de portefeuille ne peuvent pas toutes les dériver et vérifier leur utilisation sans engendrer un coût opérationnel énorme. Ainsi, ils procèdent par ordre d'index, car c'est normalement dans cet ordre que tous les logiciels de portefeuille génèrent les adresses. Le logiciel enregistre chaque adresse utilisée avant de passer à la suivante, et il cesse sa recherche lorsqu'il rencontre un nombre d'adresses consécutivement vides. Ce nombre, c'est ce que l'on appelle le Gap Limit.

Si, par exemple, le Gap Limit est fixé à `20`, et que l'adresse `m/84'/0'/0'/0/15/` est vide, le portefeuille dérivera les adresses jusqu'à `m/84'/0'/0'/0/34/`. Si cette plage d'adresses reste inutilisée, la recherche s'arrête là. Par conséquent, une transaction utilisant l'adresse `m/84'/0'/0'/0/40/` ne serait pas détectée dans cet exemple.

Termes associés : **CHEMIN DE DÉRIVATION** | **XPUB**

GATEWAY NODE

LIGHTNING NETWORK | FR : NŒUD PASSERELLE

Nœud de routage Lightning qui exploite des canaux privés (non annoncés) pour créer des opportunités de routage supplémentaires. Contrairement à un nœud de routage classique dont tous les canaux sont publics et visibles dans le graphe du réseau, un *gateway node* maintient également des canaux privés avec d'autres nœuds. Ces canaux ne sont pas diffusés via le protocole de *gossip*, et n'apparaissent donc pas dans la topologie publique du réseau.

L'intérêt d'un *gateway node* se manifeste lorsque d'autres nœuds ouvrent des canaux publics vers lui : ces connexions créent des chemins de paiement transitant par les canaux privés, offrant ainsi des routes alternatives invisibles au reste du réseau. Le *gateway node* agit alors comme un pont entre le réseau public et des nœuds qui ne souhaitent pas exposer leurs canaux.

Termes associés : **NOEUD LIGHTNING** | **PRIVATE CHANNEL**

GCS - GOLOMB-CODED SET

</> INFORMATIQUE

Structure de données probabiliste utilisée dans le BIP-0158 pour construire les *compact block filters* sur Bitcoin. Un GCS permet de tester l'appartenance d'un élément à un ensemble avec la garantie de trouver tout élément présent, mais avec une probabilité de faux positif de $1/M$.

La construction d'un GCS se déroule en plusieurs étapes : les éléments bruts sont d'abord hachés avec SipHash vers des entiers de 64 bits répartis uniformément dans l'intervalle $[0, N \times M)$. Les valeurs obtenues sont triées, puis les différences entre valeurs consécutives sont encodées par codage de Golomb-Rice. Ce codage exploite la distribution géométrique naturelle des écarts pour obtenir une compression quasi optimale : chaque différence est décomposée en un quotient (encodé en unaire) et un reste (encodé sur P bits).

Comparé aux filtres de Bloom utilisés dans le BIP-0037, le GCS produit des filtres plus compacts pour un même taux de faux positifs. En contrepartie, le test d'appartenance nécessite de décompresser séquentiellement le filtre, ce qui empêche les requêtes en temps constant. Cette structure a été retenue pour les *compact block filters* avec les paramètres $P = 19$ et $M = 784\,931$.

Termes associés : **BIP-0158** | **BLOOM FILTER**

GENÈSE

PROTOCOLE | EN : GENESIS BLOCK

Le bloc de Genèse est le premier bloc du système Bitcoin. Il incarne le lancement concret de Bitcoin. Le bloc de Genèse a été créé par le fondateur anonyme de Bitcoin, Satoshi Nakamoto, le 3 janvier 2009. Son hachage est :

```
000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
```

Ce bloc contient seulement une transaction coinbase qui génère 50 bitcoins en récompense pour le mineur (dans ce cas, Satoshi Nakamoto lui-même). Ce bloc inclut un message incorporé dans la transaction coinbase :

```
The Times 03/Jan/2009 Chancellor on brink of second bailout for
banks
```

Cette citation est une référence à un article du journal *The Times*. Le message est interprété comme une critique du système financier traditionnel et de ses dérives, ce qui a en partie motivé la création de Bitcoin en tant qu'alternative monétaire.

Puisqu'il incarne le tout premier bloc de la blockchain Bitcoin, le bloc de Genèse possède un champ contenant le hachage du bloc antérieur, mais celui-ci est rempli de zéros, car il n'existe pas de bloc précédent. Par ailleurs, les 50 bitcoins générés en récompense dans ce bloc ne sont pas dépensables au niveau protocolaire.

Termes associés : **COINBASE** | **NAKAMOTO SATOSHI**

GENESIS

▽ RGB

Dans le cadre du protocole RGB, la Genesis désigne l'ensemble des données initiales, régies par un schéma spécifique, qui établissent l'état de départ de tout contrat. Ces informations constituent le point d'ancrage à partir duquel s'opère l'évolution des smart contracts et la gestion des jetons RGB. On peut la rapprocher du concept de *Genesis Block* sur Bitcoin, ou bien au concept de transaction Coinbase, mais ici au niveau *client-side* et des jetons RGB.

Termes associés : **CONTRAT RGB** | **SCHEMA**

GETBLOCKTEMPLATE

↗ MINAGE

Protocole de communication entre un nœud Bitcoin et un logiciel de minage, défini dans le BIP-0022 et le BIP-0023. Il a été introduit en 2012 pour remplacer le protocole `getwork`, dont les limitations devenaient problématiques avec la croissance de la puissance de calcul du réseau.

Contrairement à `getwork`, qui fournissait au mineur un en-tête de bloc prêt à hacher, `getblocktemplate` transmet un modèle complet de bloc (*block template*). Le mineur reçoit la liste des transactions à inclure, les paramètres du bloc et les règles de consensus à respecter. Cette approche lui permet de construire lui-même le bloc candidat et de choisir quelles transactions y intégrer.

Ce changement a réduit le pouvoir des opérateurs de pools de minage sur la sélection des transactions, ce qui a décentralisé une partie du processus de construction des blocs. `getblocktemplate` a ensuite été largement remplacé par le protocole Stratum, puis par Stratum V2.

Termes associés : **GETWORK** | **MINING POOL**

GETWORK

↗ MINAGE

Ancien protocole de minage pour Bitcoin créé en 2010 par m0mchil. Getwork permettait aux mineurs de recevoir des données de travail de la part d'un nœud complet. Il était établi sur des requêtes RPC permettant d'obtenir des en-têtes

de blocs sur lesquels travailler pour trouver une preuve de travail valide. Getwork était optimisé pour le minage par GPU. Ce fut le premier logiciel open source conçu pour optimiser la communication entre les nœuds et les mineurs à une époque où quelques acteurs gardaient ces logiciels privés. Getwork a été progressivement remplacé par Stratum, plus efficace, notamment pour les ASICs, et moins gourmand en bande passante.

Termes associés : **MINING POOL** | **STRATUM**

GHASH.IO

↗ MINAGE

Ancienne pool de minage de Bitcoin opérée par la société CEX.IO, basée au Royaume-Uni. Ghash.io est principalement connue pour avoir brièvement dépassé 51 % du hashrate total du réseau Bitcoin en juin 2014, soulevant des craintes concrètes quant à la possibilité d'une attaque des 51 %. Bien que la pool n'ait pas exploité cette position dominante à des fins malveillantes, cet événement a constitué un signal d'alarme pour la communauté Bitcoin, illustrant les risques de centralisation liés aux pools de minage. De nombreux mineurs ont volontairement quitté Ghash.io dans les jours suivants afin de redistribuer le hashrate vers d'autres pools. Ghash.io a ensuite progressivement perdu de son importance avant de cesser ses activités.

Termes associés : **ATTAQUE DES 51%** | **MINING POOL**

GINGER WALLET

⊸ PORTEFEUILLE

Ginger Wallet est un portefeuille Bitcoin desktop open source axé sur la confidentialité. Il s'agit d'un fork de Wasabi Wallet, lancé en juin 2024 quelques jours après l'arrêt du service de coordination de coinjoins de zkSNACKs. Ginger Wallet reprend les fonctionnalités de confidentialité de Wasabi, notamment l'intégration de coinjoins WabiSabi.

Termes associés : **COINJOIN** | **TOR** | **WASABI WALLET**

GIT

‹/› INFORMATIQUE

Système de contrôle de version distribué conçu pour gérer tout type de projet logiciel avec efficacité. Il permet aux développeurs de suivre les modifications apportées au code source d'un projet au fil du temps, de revenir à des états antérieurs, de gérer des branches et de fusionner des modifications. Git facilite la collaboration entre les développeurs en permettant à plusieurs personnes de travailler sur le même projet simultanément, avec des outils pour gérer les éventuels conflits dans les fichiers. Chaque développeur travaille localement et peut ensuite synchroniser ses modifications avec le dépôt central. Créé en 2005 par Linus Torvalds, Git est devenu de fait le standard pour le contrôle de version dans l'industrie du logiciel. Les développements des implémentations de nœud Bitcoin, dont Bitcoin Core, sont gérées avec Git.

Termes associés : **BITCOIN CORE** | **GITHUB**

GITHUB

‹/› INFORMATIQUE

Plateforme de gestion et d'hébergement de code source qui facilite la collaboration entre développeurs. Comme son nom l'indique, GitHub est établi sur le système de contrôle de version Git. Cette plateforme permet donc de suivre les changements de code, de gérer les versions et d'encourager la collaboration grâce à des outils comme les pull requests et les issues. GitHub est devenu un outil incontournable pour les développeurs, notamment dans la communauté Bitcoin où la majorité des projets, y compris Bitcoin Core, l'implémentation principale du protocole, y sont hébergés. En 2018, Microsoft a acquis GitHub pour 7,5 milliards de dollars.

Termes associés : **GIT** **PULL REQUEST**

GITLAB

‹/› INFORMATIQUE

Plateforme de gestion et d'hébergement de code source qui facilite la collaboration entre développeurs. GitLab est la principale alternative à GitHub. La plateforme est assez similaire, mais elle offre également la possibilité d'être autohébergée. Comme son nom l'indique, GitLab est établi sur le système de contrôle de version Git. Cette plateforme permet donc de suivre les changements de code, de gérer les versions et d'encourager la collaboration grâce à des outils comme les merge requests et les issues. Certains projets liés à Bitcoin comme Samourai Wallet, Whirlpool ou encore RoninDojo utilisent GitLab.

Termes associés : **GIT** **GITHUB**

GLOBAL STATE

RGB

Dans le cadre du protocole RGB, le *Global State* désigne l'ensemble des propriétés publiques d'un contrat, établies lors de la *Genesis*. Il agit comme un registre public et regroupe les éléments d'état accessibles à tous et pouvant être mis à jour par des transitions autorisées conformément aux règles définies par le schéma du contrat. Contrairement aux *Owned States*, qui sont associés à des entités spécifiques, le *Global State* n'appartient à aucune partie en particulier.

Termes associés : **GENESIS** **OWNED STATE**

GNPA

CRYPTOGRAPHIE | EN : PRNG

Sigle de « Générateur de nombres pseudo-aléatoires ». Les GNPA sont une catégorie d'algorithmes utilisés pour générer des séquences de nombres approximativement aléatoires, à partir d'un état initial appelé graine (seed). En cryptographie, le GNPA est utilisé pour produire des clés, des vecteurs d'initialisation et d'autres éléments nécessitant de l'aléatoire. Un bon GNPA doit avoir des propriétés telles que l'uniformité des sorties, l'imprévisibilité et la résistance aux attaques prédictives. Contrairement aux générateurs de nombres véritablement aléatoires, les GNPA sont déterministes et reproduisibles. Sur Bitcoin, les GNPA peuvent être utilisés sur les logiciels de gestion de portefeuille ou les *hardware wallets*, afin de générer la phrase de récupération qui est à la base des portefeuilles déterministes et hiérarchiques.

Termes associés : **GRAINE** | **PHRASE DE RÉCUPÉRATION**

GNU

INFORMATIQUE

Projet initié en 1983 par Richard Stallman pour créer un système d'exploitation libre, compatible avec Unix. Le projet a développé de nombreux logiciels libres qui peuvent être utilisés comme un système d'exploitation ou en complément d'un autre système d'exploitation. GNU est à la base du mouvement du logiciel libre, largement repris dans la communauté Bitcoin. Combiné avec le noyau Linux, il forme les systèmes d'exploitation GNU/Linux.

Le nom de « GNU » est un acronyme récursif signifiant « GNU's Not Unix », que l'on peut traduire en français « GNU n'est pas Unix ».

Termes associés : **GPL** | **UNIX**

GOLDFINGER

ATTAQUE

Scénario hypothétique sur le système Bitcoin où un acteur malveillant contrôle plus de 50 % de la puissance de calcul totale du minage (*hashrate*). Avec une telle dominance, l'attaquant peut manipuler le processus de consensus, permettant des actions malveillantes telles que la double dépense, où les mêmes bitcoins sont dépensés une première fois sur une chaîne finalement rendue désuète, puis une seconde fois sur la chaîne valide. Une autre finalité d'une attaque Goldfinger est la censure des transactions. Cependant, réaliser une attaque de ce type nécessite des ressources financières, humaines, énergétiques et techniques considérables, et rend l'acteur malveillant susceptible d'être découvert avant que l'attaque n'ait lieu. Bien que théoriquement possible, une attaque Goldfinger sur Bitcoin est considérée comme très peu probable en raison de la décentralisation du minage et de la grande puissance de calcul actuellement déployée.

Cette attaque est également nommée « Attaque des 51 % ».

Termes associés : **DOUBLE DÉPENSE** | **HASHRATE**

GOSSIP

LIGHTNING NETWORK | FR : BAVARDAGE

Gossip désigne un algorithme distribué pair-à-pair (P2P) pour diffuser l'information de manière épidémique à tous les agents du réseau. Pour Bitcoin, Lightning et d'autres systèmes distribués, ce protocole permet d'échanger et de synchroniser l'état global des nœuds en peu de cycles. Chaque nœud propage une information à un ou plusieurs voisins choisis aléatoirement ou non, ces derniers, à leur tour, propagent l'information à d'autres voisins et ainsi de suite jusqu'à arriver à un état synchronisé globalement.

Dans le cadre de Lightning, le gossip est un protocole de communication entre les nœuds pour partager les informations sur l'état actuel et la topologie du réseau. Le protocole de gossip permet aux nœuds de connaître l'état dynamique des canaux de paiement et des autres nœuds, afin de faciliter le routage des transactions à travers le réseau sans nécessiter de connexions directes entre tous les nœuds.

Termes associés : **NOEUD** | **NOEUD LIGHTNING**

GPL

INFORMATIQUE

Sigle de « *GNU General Public License* ». C'est une série de licences de logiciel libre initialement créée par la Free Software Foundation. Elle garantit aux utilisateurs la liberté d'utiliser, de modifier et de distribuer le logiciel. La GPL exige toutefois que tout logiciel dérivé ou modifié soit également distribué sous la même licence, ce qui permet d'assurer que les libertés initiales sont préservées dans toutes les versions du logiciel. La GPL est souvent utilisée pour les projets open source et on la retrouve parfois sur certains projets liés à Bitcoin. Toutefois, la licence initiale du client Bitcoin de Satoshi Nakamoto était une MIT X11. C'est d'ailleurs toujours le cas pour Bitcoin Core.

Termes associés : **FOSS** | **GNU**

GRAFTROOT

PROTOCOLE

Proposition d'amélioration du protocole Bitcoin formulée par Gregory Maxwell en février 2018. Graftroot étend le concept de Taproot en y ajoutant un mécanisme de délégation des conditions de dépense. Avec Taproot, un groupe de participants qui partagent une clé agrégée peut soit dépenser directement avec cette clé, soit révéler un script alternatif contenu dans un arbre de Merkle. Graftroot permet en plus à ces participants de déléguer leur capacité de dépense à des scripts arbitraires en signant ces scripts avec leur clé agrégée.

L'utilité principale de ce protocole réside dans l'absence de limite sur le nombre de conditions de dépense alternatives. Chaque délégation se matérialise par une simple signature Schnorr qui porte sur le script autorisé, sans surcoût proportionnel au nombre de scripts délégués. Dans Taproot, l'ajout de branches dans l'arbre de Merkle accroît la taille de la preuve de dépense, tandis que Graftroot maintient un coût constant.

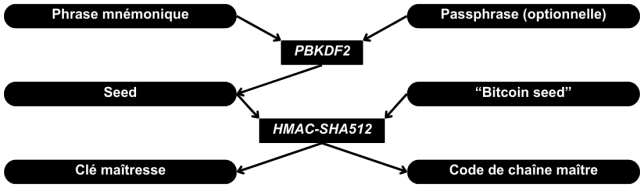
Graftroot n'a jamais été formalisé sous la forme d'un BIP et n'a pas été intégré au protocole Bitcoin.

Termes associés : **SCHNORR** | **TAPROOT**

GRAINE

PORTEFEUILLE | EN : SEED

Dans le cadre spécifique d'un portefeuille déterministe hiérarchique Bitcoin, une graine est une information de 512 bits issue d'un aléa. Elle permet de générer de manière déterministe et hiérarchique un ensemble de clés privées, et leurs clés publiques correspondantes, pour un portefeuille Bitcoin. La graine est souvent confondue avec la phrase de récupération en elle-même. Pourtant, c'est une information différente. La graine est obtenue en appliquant la fonction `PBKDF2` sur la phrase mnémonique et sur l'éventuelle passphrase.



La graine a été inventée avec le BIP-0032 qui définit les bases du portefeuille déterministe hiérarchique. Dans ce standard, la graine peut mesurer entre 128 et 512 bits (256 bits étant conseillés). Cela permet de dériver toutes les clés d'un portefeuille depuis une information unique, contrairement aux portefeuilles JBOK (*Just a Bunch Of Keys*) qui nécessitent de réaliser de nouvelles sauvegardes pour toute clé générée. Le BIP-0039 est par la suite venu proposer un encodage de cette graine, afin de simplifier sa lecture par l'humain. Cet encodage se fait sous la forme d'une phrase, que l'on nomme généralement phrase mnémonique ou phrase de récupération. Ce standard permet d'éviter les erreurs au niveau de la sauvegarde de la graine, notamment grâce à l'utilisation d'une somme de contrôle.

Hors du contexte de Bitcoin, dans le domaine de la cryptographie en général, une seed est une valeur initiale utilisée pour générer des clés cryptographiques, ou plus largement, n'importe quel type de données produites par un générateur de nombres pseudo-aléatoires. Cette valeur initiale doit être aléatoire et imprévisible pour garantir la sécurité du système cryptographique. En effet, la seed introduit de l'entropie dans le système, mais le processus de génération qui suit est lui déterministe.

Dans le langage courant, la majorité des bitcoiners se réfèrent à la phrase mnémonique quand ils parlent de la « seed », et non à l'état intermédiaire de dérivation qui se situe entre la phrase mnémonique et la clé maîtresse.

Termes associés : **BIP-0039** | **PHRASE DE RÉCUPÉRATION**

GREEN ADDRESS

PORTEFEUILLE

Ancien logiciel de portefeuille Bitcoin racheté en juillet 2016 par Blockstream pour créer le logiciel actuel Green Wallet.

Termes associés : **BLOCKSTREAM** **GREEN WALLET**

GREEN ADDRESSES

PORTEFEUILLE

Vieille proposition d'adresse Bitcoin dont les transactions qui en sont émises sont pré-approuvées par un tiers de confiance. Ce concept permet aux parties recevant des fonds via cette adresse de les considérer comme immédiatement fiables, sans attendre qu'elles soient incluses dans un bloc. Cela repose sur la confiance accordée au service qui contrôle la *green address*, car il garantit que les fonds envoyés n'ont pas été et ne seront pas double dépensés. Ce type de service était autrefois utilisé pour accélérer les transactions (zero-conf), mais il n'est plus utilisé de nos jours.

Termes associés : **DOUBLE DÉPENSE** **ZEROCONF**

GREEN WALLET

PORTEFEUILLE

Logiciel de portefeuille Bitcoin disponible sur PC, Android et iOS développé par Blockstream depuis l'acquisition du logiciel GreenAddress en 2016. Il intègre plusieurs fonctionnalités comme une protection multisignatures avec une authentification à deux facteurs. Il est également compatible avec la plupart des hardware wallets. C'est un logiciel simple à prendre en main qui peut être utilisé par des débutants.

Termes associés : **BLOCKSTREAM** **GREEN ADDRESS**

GRIEFING ATTACK

ATTAQUE

Attaque sur le Lightning Network dans laquelle un acteur malveillant sature un ou plusieurs canaux de paiement avec des HTLC qui ne se résolvent jamais, en refusant de révéler la préimage de paiement. Les fonds des victimes se retrouvent gelés pendant toute la durée du timelock, sans que l'attaquant ne vole quoi que ce soit.

Chaque canal ne peut contenir que 483 HTLC en cours par direction. Un attaquant peut donc bloquer entièrement un canal avec un nombre relativement faible de paiements fictifs. La seule option pour la victime est de forcer la fermeture du canal, ce qui engendre des frais on-chain élevés puisque chaque HTLC en attente génère une transaction supplémentaire sur la chaîne.

Bien que l'attaquant n'obtienne pas de gain monétaire direct, il peut tirer un bénéfice indirect en paralysant les routes concurrentes pour capter davantage de frais de routage, ou en exigeant une rançon pour cesser l'attaque. Cette vulnérabilité reste un problème ouvert. Les contre-mesures proposées incluent le protocole HTLC-GP, qui impose à chaque participant de verrouiller une pé-

nalité financière récupérable par les victimes en cas de griefing, ainsi que des mécanismes de paiements anticipés (upfront payments).

Termes associés : **CANAL DE PAIMENT** | **LIGHTNING NETWORK**

GROTH16

CRYPTOGRAPHIE

Système de preuve à divulgation nulle de connaissance de type SNARK, proposé par le cryptographe Jens Groth en 2016. Groth16 se distingue par la taille minimale de ses preuves (seulement trois éléments de courbe elliptique, soit environ 200 octets) et par un temps de vérification constant, ce qui en fait l'un des schémas les plus efficaces pour la vérification on-chain. En contrepartie, Groth16 nécessite une cérémonie de configuration de confiance (*trusted setup*) spécifique à chaque circuit, durant laquelle des paramètres secrets sont générés puis idéalement détruits. Si ces paramètres étaient reconstitués, il serait possible de créer de fausses preuves. Le framework RISC Zero utilise Groth16 comme couche d'encapsulation finale pour compresser les preuves STARK volumineuses en preuves SNARK compactes. C'est ce mécanisme qui est exploité par le rollup Citrea pour publier des preuves de validité sur la blockchain Bitcoin.

Termes associés : **RISC ZERO** | **SNARK**

GRPC

INFORMATIQUE

Framework d'appels de procédures distantes (*Remote Procedure Call*) développé par Google. gRPC utilise le format Protocol Buffers pour sérialiser les données et le protocole HTTP/2 pour le transport, offrant des communications performantes et typées entre un client et un serveur.

Dans l'écosystème Lightning, gRPC est l'interface principale utilisée par LND pour exposer ses fonctionnalités. Les outils comme LNCLI interagissent avec le nœud Lightning via des appels gRPC pour gérer les canaux, envoyer des paiements, consulter les balances ou configurer le routage.

Termes associés : **LNCLI** | **LND**

GUI

INFORMATIQUE | FR : INTERFACE GRAPHIQUE UTILISATEUR

Acronyme de « *Graphical User Interface* », ou « interface graphique utilisateur » en français. C'est une forme d'interface utilisateur qui permet d'interagir avec des logiciels à travers des éléments visuels interactifs (boutons, icônes, images, fenêtres...) et qui privilégie l'utilisation de dispositifs de pointage (la souris) plutôt que de commandes textuelles comme avec la CLI.

Termes associés : **BITCOIN CORE** | **CLI**

GUISETTINGS.INI.BAK

PROTOCOLE

Fichier dans Bitcoin Core utilisé pour stocker une sauvegarde des paramètres de l'interface graphique (GUI). Cette sauvegarde est créée lors de l'utilisation de l'option `-resetguisettings`, qui réinitialise les paramètres de la GUI à leurs valeurs par défaut. Ce fichier permet à l'utilisateur d'inspecter rétroactivement ses anciennes configurations afin de diagnostiquer d'éventuels problèmes liés aux paramètres de la GUI.

Termes associés : **BITCOIN CORE** **GUI**

H

HACHEUR

↗ MINAGE | EN : HASHER

Dans le contexte des pools de minage, les participants sont souvent désignés sous le terme de « hacheurs ». Ces mineurs individuels ont pour tâche principale de hacher des blocs *templates* fournis par le serveur de la pool, en recherchant des hachages qui satisfont la cible de difficulté définie pour les *shares*, et non celle de Bitcoin. Le reste du processus de minage, qui inclut la construction effective des blocs, la sélection des transactions ou la vérification et la diffusion de la preuve de travail valide selon la difficulté propre à Bitcoin, est effectué directement par les pools.

Terme associé : **MINING POOL**

HALVING

✿ PROTOCOLE

Le terme « halving » (division par deux) fait référence à un événement programmé qui réduit de moitié la récompense attribuée aux mineurs pour chaque bloc miné via la subvention de bloc. Cette réduction s'applique spécifiquement à la partie de la subvention de bloc constituée de nouveaux bitcoins créés ex-nihilo. Le halving a été conçu par Satoshi Nakamoto, le créateur de Bitcoin, comme un mécanisme permettant de contrôler l'inflation et d'assurer un approvisionnement limité en bitcoins.

La subvention de bloc initiale était de 50 bitcoins, et le halving se produit tous les 210 000 blocs minés, ce qui prend environ quatre ans. Les halvings ont eu lieu :

- Au bloc 210 000 le 28 novembre 2012 pour réduire la subvention à 25 BTC ;
- Au bloc 420 000 le 9 juillet 2016 pour réduire la subvention à 12,5 BTC ;
- Au bloc 630 000 le 11 mai 2020 pour réduire la subvention à 6,25 BTC ;
- Au bloc 840 000 le 20 avril 2024 pour réduire la subvention à 3,125 BTC.

Les halvings continueront à se produire jusqu'à ce que la subvention de bloc atteigne zéro, moment auquel l'offre maximale d'environ 21 millions de bitcoins aura été atteinte. Le prochain halving de Bitcoin devrait avoir lieu aux alentours du printemps 2028, bien que la date exacte puisse légèrement varier en fonction du temps de minage des blocs. À ce moment-là, la subvention de bloc sera réduite de 3,125 à 1,5625 bitcoins. Ce sera le cinquième halving de l'histoire de Bitcoin. À mesure que les subventions de bloc baissent, les frais de transaction deviennent une source de revenus de plus en plus importante pour les mineurs, ce qui garantit leur motivation à continuer à participer à la preuve de travail.

Termes associés : **RÉCOMPENSE DE BLOC** | **SUBVENTION DE BLOC**

HARD FORK

✿ PROTOCOLE

Modification des règles du protocole de manière non rétrocompatible. Cette modification donne lieu à une séparation définitive du réseau de nœuds Bitcoin en deux groupes distincts : les nœuds avec la mise à jour et les nœuds sans la mise à jour. Cette scission se matérialise par la division de la blockchain originale en

deux blockchains distinctes, partageant toutefois un historique commun, d'où l'usage du terme « fork », traduisible en français par « embranchement, bifurcation ».

Une modification est dite non rétrocompatible lorsqu'elle est soit extensive, c'est-à-dire qu'elle supprime ou rend moins restrictives certaines règles du protocole, soit bilatérale, c'est-à-dire qu'elle ajoute des règles incompatibles avec les anciennes. En d'autres termes, un hard fork s'observe lorsque certains nœuds font en sorte qu'un bloc invalide devienne valide. En résulte alors la formation d'une nouvelle version du protocole, qui peut soit remplacer le Bitcoin original si une majorité est trouvée, soit devenir un altcoin indépendant s'il n'est qu'utilisé en marge. Par exemple, Bitcoin Cash (BCH) est un hard fork de Bitcoin. L'embranchement a eu lieu au bloc n° 478 559, le 1er août 2017.

Termes associés : **BLOCKCHAIN** | **SOFT FORK**

HARDWARE WALLET

PORTEFEUILLE | FR : PORTEFEUILLE MATÉRIEL

Un hardware wallet, ou portefeuille matériel, est un dispositif électronique dédié à la sécurisation et à la gestion des clés privées d'un portefeuille Bitcoin. Ces périphériques sont conçus pour procurer une sécurité renforcée par rapport aux portefeuilles logiciels qui sont sur des machines polyvalentes et directement connectées à internet. Les hardware wallets stockent la phrase mnémonique hors ligne, sur un matériel qui dispose d'une infime surface d'attaque, ce qui l'isole des environnements potentiellement vulnérables. Lorsqu'une transaction est effectuée, le portefeuille matériel signe la transaction à l'intérieur du dispositif lui-même, sans exposer la clé privée à l'extérieur. Une fois la transaction signée, elle est transmise au réseau Bitcoin pour être confirmée et incluse dans la blockchain Bitcoin. Parmi les modèles de hardware wallets les plus populaires, on peut citer : Ledger, Trezor, Coldcard, Passport, BitBox, Satochip, Jade ou encore SeedSigner (liste non exhaustive).

Certains bitcoiners préfèrent que l'on emploie le terme de « périphérique de signature », ou « signing device » en anglais, afin d'éviter de faire penser que les bitcoins se trouvent physiquement dans le portefeuille.

Termes associés : **CLÉ PRIVÉE** | **PORTEFEUILLE**

HASH160

CRYPTOGRAPHIE

Fonction cryptographique utilisée sur Bitcoin notamment pour générer des adresses de réception Legacy et SegWit v0. Elle combine deux fonctions de hachage qui s'exécutent successivement sur l'input : d'abord SHA256, puis RIPEMD160. La sortie de cette fonction est donc de 160 bits.

$$HASH160(x) = RIPEMD160(SHA256(x))$$

Termes associés : **RIPEMD160** | **SHA256**

HASH256

CRYPTOGRAPHIE

Fonction cryptographique utilisée pour diverses applications sur Bitcoin. Elle consiste en l'application double de la fonction SHA256 sur les données en entrée. Le message est passé une première fois dans SHA256, et le résultat de cette opération est utilisé comme entrée pour passer une seconde fois dans SHA256. La sortie de cette fonction est donc de 256 bits.

$$HASH256(x) = SHA256(SHA256(x))$$

Termes associés : **FONCTION DE HACHAGE** | **SHA256**

HASHBOARD

MINAGE

Carte électronique qui constitue le composant principal d'un mineur ASIC. Chaque hashboard contient les puces de hachage (les ASIC à proprement parler) qui effectuent les calculs `SHA256` nécessaires au minage de Bitcoin. Un mineur ASIC typique, comme un Antminer S9 ou S19, intègre généralement trois hashboards reliés à une carte de contrôle.

Terme associé : **ASIC**

HASHCASH

HISTOIRE

Système de preuve de travail conçu par Adam Back en 1997 pour lutter contre le spam et les attaques DoS. Il repose sur le principe qu'un expéditeur doit effectuer un travail de calcul (spécifiquement, la recherche d'une collision partielle sur une fonction de hachage cryptographique) pour prouver son travail. Cette tâche est coûteuse en temps et en énergie pour l'expéditeur, mais la vérification du résultat par le destinataire est rapide et simple. Ce protocole s'est révélé particulièrement adapté à la lutte contre le spam dans les messageries électroniques, car il est peu contraignant pour les utilisateurs légitimes, tout en constituant un obstacle majeur pour les spammeurs. En effet, l'envoi d'un seul courriel requiert quelques secondes de calcul, mais reproduire cette opération des millions de fois rend l'opération extrêmement coûteuse en termes d'énergie et de temps, ce qui vient souvent annuler l'intérêt économique des campagnes de spam, qu'elles soient à but marketing ou malveillant. De plus, il permet de préserver l'anonymat de l'expéditeur.

HashCash a rapidement été adopté par des cypherpunks qui cherchaient à développer un système de monnaie électronique anonyme sans intermédiaire. En effet, l'innovation d'Adam Back a introduit pour la première fois la notion de rareté dans le monde numérique. On retrouve alors le concept de preuve de travail dans plusieurs systèmes de monnaies électroniques antérieurs à Bitcoin, dont :

- b-money de Wei Dai publié en 1998 ;
- RPOW de Hal Finney publié en 2004 ;
- Bit Gold de Nick Szabo publié en 2005.

Le principe de HashCash se retrouve également au sein du protocole Bitcoin, où il est utilisé comme mécanisme de protection face aux attaques Sybil.

Termes associés : **B-MONEY** | **PREUVE DE TRAVAIL**

HASHRATE

↗ MINAGE | FR : TAUX DE HACHAGE

Indicateur de la puissance de calcul du réseau, mesurée en hachages par seconde (H/s). Il indique la capacité des mineurs à exécuter des opérations de hachage dans le cadre de la preuve de travail. Un hashrate élevé signifie une plus grande sécurité de l'historique économique de Bitcoin et une plus grande résistance aux attaques, car il faudrait une importante quantité de puissance de calcul pour compromettre le réseau.

Le hashrate est également indicatif de la concurrence entre les mineurs : plus le hashrate est élevé, plus la difficulté de minage est grande, ce qui influence la répartition des récompenses, et donc la rentabilité des mineurs. C'est donc un indicateur clé de la santé et de la sécurité du système Bitcoin. De la même manière que le hashrate sert à mesurer la puissance de calcul globale du réseau Bitcoin, il peut également être utilisé pour mesurer la puissance de calcul d'une machine, d'une ferme de minage ou encore d'une pool de minage.

Termes associés : **DIFFICULTÉ** | **PREUVE DE TRAVAIL**

HASHRATE HIJACKING

✸ ATTAQUE | FR : DÉTOURNEMENT DE HASHRATE

Attaque qui consiste à détourner la puissance de calcul d'un mineur vers une destination contrôlée par l'attaquant, sans que le mineur en soit conscient. Cette attaque est possible lorsque la communication entre le mineur et le serveur de la pool de minage n'est pas chiffrée. Un attaquant positionné entre les deux (attaque de l'homme du milieu) peut intercepter les données échangées et remplacer l'adresse de la pool légitime par la sienne, redirigeant ainsi le hashrate du mineur à son profit.

Le hashrate hijacking est facilité par le fait que le protocole Stratum V1 transmet les données en clair, sans authentification ni chiffrement. Cette vulnérabilité a motivé le développement de Stratum V2, qui intègre un chiffrement AEAD pour sécuriser les communications entre les mineurs et les pools, rendant ce type d'attaque beaucoup plus difficile.

Termes associés : **HASHRATE** | **MINING POOL**

HAUTEUR DE BLOC

PROTOCOLE | EN : BLOCK HEIGHT

Désigne le numéro de séquence d'un bloc particulier par rapport au premier bloc, connu sous le nom de « bloc de Genèse », qui est indexé à la hauteur zéro. Cet indicateur nous donne le nombre de blocs qui précèdent le bloc étudié. Par exemple, si un bloc est à la hauteur 650 000, cela signifie qu'il y a 650 000 blocs qui le précèdent. La hauteur de bloc est utilisée pour identifier un bloc spécifique au sein de la blockchain. Contrairement à une croyance répandue, la hauteur de bloc ne détermine pas quelle est la chaîne valide, car les nœuds se synchronisent sur la chaîne ayant accumulé le plus de travail, et non sur la chaîne la plus longue.

Termes associés : **BLOC** | **BLOCKCHAIN**

HD - HIERARCHICAL-DETERMINISTIC

PORTEFEUILLE | FR : DÉTERMINISTE ET HIÉRARCHIQUE

Se dit d'un portefeuille Bitcoin qui utilise une information unique (la graine ou « seed » en anglais) pour générer une multitude de paires de clés publiques et privées de manière séquentielle et reproductible. Cette manière de gérer des clés est définie par le standard BIP-0032. L'avantage principal des portefeuilles HD est qu'ils permettent aux utilisateurs de disposer d'une multitude de paires de clés différentes, notamment afin d'éviter la réutilisation d'adresse, tout en pouvant toutes les régénérer depuis une information unique. On dit de cette structure qu'elle est hiérarchique, car elle permet de créer une organisation en arborescence de multiples clés et adresses à partir d'une seule graine. Et elle est déterministe dans le sens où chaque graine génère toujours la même séquence de clés dans n'importe quel portefeuille conforme à ce système.

Termes associés : **BIP-0032** | **BIP-0044**

HEADERS-FIRST SYNC

RÉSEAU | FR : SYNCHRONISATION ENTÊTES D'ABORD

Méthode de synchronisation initiale (IBD) introduite dans Bitcoin Core 0.10.0, qui consiste à télécharger d'abord les en-têtes de tous les blocs avant de télécharger les blocs complets eux-mêmes. Cette méthode a remplacé la synchronisation *blocks-first*, utilisée par les versions antérieures de Bitcoin Core, qui téléchargeait les blocs séquentiellement depuis un seul pair.

Termes associés : **IBD - INITIAL BLOCK DOWNLOAD** | **NOEUD COMPLET**

HEURE UNIX

</> INFORMATIQUE | EN : UNIX TIME

L'Heure Unix ou Temps Unix représente le nombre de secondes écoulées depuis le 1er janvier 1970 à minuit UTC (Époque Unix). Ce système est utilisé dans les systèmes d'exploitation Unix et dérivés pour marquer le temps de manière universelle et standardisée. Il permet la synchronisation des horloges et la gestion des événements dans le temps, indépendamment des fuseaux horaires.

Dans le cadre de Bitcoin, on l'utilise pour l'horloge locale des nœuds, et donc pour le calcul du NAT. Le network-adjusted time est une médiane du temps des nœuds calculée en local par chaque nœud, et ce référentiel est ensuite utilisé pour vérifier la validité des horodatages des blocs. En effet, pour qu'un bloc soit accepté par un nœud, son horodatage doit se situer entre le MTP (temps médian des 11 derniers blocs minés) et le NAT plus 2 heures :

```
MTP < Horodatage valide < (NAT + 2h)
```

On utilise également l'Heure Unix pour établir des *timelocks*, lorsque ceux-ci se basent sur l'heure réelle, et non pas sur un nombre de blocs.

Termes associés : **HORODATAGE** | **TIMELOCK**

HEURISTIQUE D'ANALYSE

CONFIDENTIALITÉ | EN : ANALYSIS HEURISTIC

Une heuristique d'analyse de chaîne sur Bitcoin est une famille de méthodes empiriques utilisées pour tracer les flux de bitcoins sur la blockchain en se basant sur des caractéristiques observées dans les transactions. Une heuristique est une approche pratique qui permet de résoudre des problèmes, souvent par des méthodes approximatives, mais qui représente une solution suffisamment bonne pour atteindre un objectif donné. Ces heuristiques permettent d'obtenir des résultats assez fiables, mais jamais d'une précision absolue. En d'autres termes, l'analyse de chaîne implique toujours une dimension de vraisemblance dans les conclusions émises. Par exemple, on pourra estimer avec plus ou moins de certitude que deux adresses appartiennent à une même entité, mais une certitude totale sera toujours hors de portée. Tout l'objectif de l'analyse de chaîne réside précisément dans l'agrégation de diverses heuristiques en vue de minimiser le risque d'erreur. Il s'agit en quelque sorte d'une accumulation de preuves qui nous permet de nous approcher davantage de la réalité. Dans ce cadre, on différencie les heuristiques internes et les heuristiques externes.

Les heuristiques internes se concentrent sur les caractéristiques spécifiques à l'intérieur d'une transaction individuelle. Elles incluent dans leur analyse des éléments tels que les montants des UTXOs, les scripts utilisés, les versions ou encore les *locktimes*. Par exemple, l'heuristique du paiement rond permet d'identifier une sortie de transaction comme étant vraisemblablement un paiement si son montant est un nombre rond. Ces heuristiques permettent souvent d'identifier le change (rendu de monnaie qui revient vers le même utilisateur) et donc de continuer le traçage.

Les heuristiques externes, quant à elles, analysent les similitudes et les caractéristiques au-delà de la transaction en elle-même. Elles englobent tout l'environnement de la transaction. Par exemple, la réutilisation d'adresse sur plusieurs transactions est une heuristique externe. La CIOH en est également une.

Termes associés : **ANALYSE DE CHAINE** | **CIOH**

HEXADÉCIMAL

</> INFORMATIQUE

Système de numération en base 16, qui utilise donc 16 symboles distincts pour représenter les nombres. Les 10 premiers symboles sont les chiffres de 0 à 9, identiques à ceux du système décimal (base 10), et les six symboles suivants sont représentés par les lettres A à F. Ainsi, A représente le nombre 10, B représente 11, jusqu'à F qui représente 15. Ce système est particulièrement utilisé en informatique, car il offre une représentation plus concise des nombres binaires (base 2), et chaque chiffre hexadécimal représente exactement 4 bits, ce qui simplifie les conversions. On retrouve très souvent ce système de numération pour représenter les informations sur Bitcoin.

Termes associés : **BASE - ARITHMÉTIQUE** | **BINAIRE**

HMAC-SHA512

CRYPTOGRAPHIE

`HMAC-SHA512` est l'acronyme de « *Hash-based Message Authentication Code - Secure Hash Algorithm 512* ». C'est un algorithme cryptographique utilisé pour vérifier l'intégrité et l'authenticité des messages échangés entre deux parties. Il combine la fonction de hachage cryptographique `SHA512` avec une clé secrète partagée pour générer un code d'authentification de message (MAC) unique pour chaque message.

Dans le contexte de Bitcoin, l'utilisation naturelle de `HMAC-SHA512` est légèrement dérivée. On utilise cet algorithme dans le processus de dérivation déterministe et hiérarchique de l'arbre de clés cryptographiques d'un portefeuille. `HMAC-SHA512` est notamment utilisé pour passer de la graine (seed) à la clé maîtresse, puis pour chaque dérivation d'une paire parent vers des paires enfants. On retrouve également cet algorithme au sein d'un autre algorithme de dérivation, nommé `PBKDF2`, utilisé pour passer de la phrase de récupération et de la passphrase à la graine.

Termes associés : **PBKDF2** | **SHA512**

HODL

ÉCONOMIE ET RÉGULATION

Terme populaire dans la communauté Bitcoin qui désigne le fait de conserver ses bitcoins sur le long terme, malgré la volatilité des marchés, et de ne pas les vendre. Le terme est né d'une faute de frappe dans un message posté en 2013 sur le forum BitcoinTalk par l'utilisateur *GameKyuubi* qui semblait être en état d'ébriété, dans lequel il écrit « I AM HODLING » au lieu de « I AM HOLDING », ce qui signifie « garder » en anglais. Ce mot est rapidement devenu un mème et un slogan.

Termes associés : **BEAR MARKET** | **BITCOINTALK**

HODL HODL

ORGANISATION

Plateforme d'échange de bitcoins en pair-à-pair (P2P) qui permet aux utilisateurs d'acheter et de vendre des bitcoins directement entre eux, sans intermédiaire custodial. Fondée en 2016, Hodl Hodl se distingue des plateformes d'échange centralisées par le fait qu'elle ne détient jamais les fonds des utilisateurs. Les transactions sont sécurisées grâce à un système d'*escrow* multisig 2-de-3 : les bitcoins sont verrouillés dans une adresse multisignature associée à trois clés (acheteur, vendeur et Hodl Hodl), dont deux suffisent pour libérer les fonds. Ce mécanisme réduit le risque de contrepartie tout en permettant des échanges sans procédure KYC obligatoire. Hodl Hodl organise également la « Baltic Honeybadger Conference », un événement annuel dédié à Bitcoin.

Termes associés : **MULTISIG** | **PEACH**

HODL INVOICE

LIGHTNING NETWORK

Type particulier de facture Lightning dans lequel le destinataire reçoit le paiement mais ne révèle pas immédiatement la *preimage* au réseau. Le paiement reste ainsi « suspendu » : les fonds sont bloqués dans les HTLC le long de la route, sans que le destinataire ne les encaisse définitivement. Le destinataire peut ensuite choisir de régler la facture en libérant la *preimage*, ou de l'annuler pour rembourser automatiquement l'expéditeur.

Les *hodl invoices* sont utiles pour les dépôts remboursables, les enchères en ligne, les systèmes d'escrow ou tout mécanisme nécessitant une confirmation conditionnelle avant le règlement final. Elles exploitent la nature même des HTLC : tant que le secret n'est pas révélé, le paiement peut être annulé sans frais.

Termes associés : **HTLC** | **PREIMAGE**

HOME MINING

↗ MINAGE | FR : MINAGE À DOMICILE

Pratique qui consiste à exploiter, en tant que particulier, un ou plusieurs ASIC dans un environnement résidentiel plutôt que dans un centre de données industriel. Certains fabricants proposent des appareils spécifiquement conçus pour un usage domestique, avec des niveaux sonores réduits et des formats compacts. Des solutions de réutilisation de la chaleur se développent également : des ASIC intégrés dans des radiateurs ou des chauffe-eau permettent de réutiliser la chaleur produite par le minage pour le chauffage domestique, ce qui compense le coût énergétique.

Termes associés : **ASIC** | **SOLO MINING**

HONG-KONG ROUNDTABLE

HISTOIRE

Évènement historique de la Blocksize War qui s'est tenu le 20 février 2016 à Hong-Kong. C'était une réunion importante entre les développeurs de Bitcoin Core et les mineurs pour discuter de l'évolutivité du système et de la stratégie à adopter pour le faire passer à l'échelle. Les tensions étaient élevées avant la réunion, notamment à cause de la montée en puissance de Bitcoin Classic, une proposition de *hard fork* soutenue par Gavin Andresen qui visait à augmenter la taille des blocs à 2 Mo. Lors de cette réunion, des personnalités influentes dans l'écosystème ont pris part aux débats, notamment Jihan Wu et Micree Zhan, les co-fondateurs de Bitmain, Adam Back, le président de Blockstream, ou encore Luke Dashjr, Matt Corallo et Peter Todd.

Les mineurs, frustrés par le manque de progrès, ont menacé de soutenir Bitcoin Classic si un *hard fork* à 2 Mo n'était pas implémenté. Les développeurs ont défendu SegWit comme une solution viable. Après des heures de négociations tendues, un accord a été trouvé, stipulant que les développeurs de Bitcoin Core travailleraient sur un *hard fork* après l'implémentation de SegWit. Cet accord était censé apaiser les tensions, mais il a finalement créé davantage de méfiance et de confusion. Chaque camp a interprété l'accord différemment, ce qui a exacerbé les divisions au sein de la communauté. Bien que cet accord ait temporairement freiné l'élan de Bitcoin Classic, il a été perçu par beaucoup comme une solution insatisfaisante et maladroite.

Terme associé : **BLOCKSIZE WAR**

HORODATAGE

CRYPTOGRAPHIE | EN : TIMESTAMP

L'horodatage, ou « *timestamp* » en anglais, est un mécanisme qui consiste à associer un repère temporel précis à un événement, une donnée ou un message. Dans le contexte généraliste des systèmes informatiques, l'horodatage sert à déterminer l'ordre chronologique des opérations et à vérifier l'intégrité des données en fonction du temps.

Dans le cas spécifique de Bitcoin, les horodatages permettent d'établir la chronologie des transactions et des blocs. Chaque bloc dans la blockchain contient un horodatage indiquant le moment approximatif de sa création. Satoshi Nakamoto parle même d'un « serveur d'horodatage », dans son *White Paper*, pour décrire ce que l'on appellerait aujourd'hui la « blockchain ». Le rôle de l'horodatage sur Bitcoin est de déterminer la chronologie des transactions, afin de pouvoir déterminer, sans l'intervention d'une autorité centrale, quelle transaction est arrivée en premier. Ce mécanisme permet à chaque utilisateur de vérifier la non-existence d'une transaction par le passé, et donc d'éviter qu'un utilisateur malintentionné opère une double dépense. Ce mécanisme est justifié par Satoshi Nakamoto dans son *White Paper* par la célèbre phrase : « *Le seul moyen pour confirmer l'absence d'une transaction est d'être au courant de toutes les transactions.* » Cette norme est établie sur l'heure Unix, qui représente le total de secondes passées depuis le premier janvier 1970.

L'horodatage des blocs est relativement flexible sur Bitcoin, car pour qu'un horodatage soit considéré comme valide, il est simplement nécessaire qu'il soit plus grand que le temps médian des 11 blocs qui le précèdent (MTP) et plus petit que la médiane des temps retournés par les nœuds (network-adjusted time) plus 2 heures.

Termes associés : **HEURE UNIX** | **MTP - MEDIAN TIME PAST**

HOSTED CHANNEL

LIGHTNING NETWORK | FR : CANAL HÉBERGÉ

Canal Lightning qui n'est pas ancré dans une transaction on-chain, contrairement aux canaux classiques. Un *hosted channel*, aussi appelé « canal custodial » ou « canal virtuel », ne repose sur aucune transaction de financement sur la blockchain Bitcoin : il n'existe qu'au niveau du protocole Lightning, sans garantie de règlement on-chain.

Le fonctionnement implique deux rôles asymétriques : un hôte, qui fournit le canal avec de la liquidité de son côté, et un client, qui utilise ce canal pour accéder au réseau Lightning. Les fonds restent sous la garde de l'hôte, ce qui rend le modèle custodial. Toutefois, par rapport aux portefeuilles custodials traditionnels, les *hosted channels* offrent des garanties supplémentaires : le client génère lui-même ses factures et contrôle les préimages de paiement, ce qui rend la fraude détectable. L'hôte et le client échangent des données d'état signées mutuellement, ce qui fournit une preuve cryptographique du solde dû. L'ouverture et la fermeture de ces canaux ne coûtent rien en frais de minage, ce qui facilite l'intégration de nouveaux utilisateurs sur le Lightning Network à

moindre coût.

Termes associés : **CANAL DE PAIMENT** | **TURBO CHANNEL**

HRP - HUMAN READABLE PART

‹/› INFORMATIQUE

Le HRP, pour « *Human Readable Part* » (partie lisible par l'homme), est un composant des adresses de réception bech32 et bech32m (SegWit v0 et SegWit v1). Le HRP fait référence à la portion de l'adresse qui est spécifiquement formatée pour être facilement lue et interprétée par les humains. Prenons l'exemple d'une adresse Bitcoin bech32 :

```
bc1qar0srrr7xfkvy5l643lydnw9re59gtzzwf5mdq
```

Dans cette adresse, le `bc` initial représente le HRP. Ce préfixe permet d'identifier en un coup d'œil que cette suite de caractères que l'on nous présente est une adresse Bitcoin et pas autre chose.

Termes associés : **ADRESSE DE RÉCEPTION** | **BECH32 ET BECH32M**

HSM

CRYPTOGRAPHIE | FR : MODULE DE SÉCURITÉ MATÉRIEL

Sigle de « Hardware Security Module » (module de sécurité matériel). Il s'agit d'un dispositif physique dédié à la gestion sécurisée de clés cryptographiques. Un HSM effectue les opérations de génération de clés, de signature et de chiffrement dans un environnement matériel inviolable, conçu pour résister aux tentatives d'extraction physique ou logique des secrets qu'il contient.

Dans le contexte de Bitcoin, les HSM sont principalement utilisés par les entreprises et les plateformes d'échange qui gèrent d'importants volumes de fonds. Ils permettent de signer des transactions Bitcoin de manière sécurisée sans que les clés privées ne quittent jamais le périmètre protégé du module. Contrairement aux *hardware wallets* destinés aux particuliers, les HSM sont des équipements de qualité industrielle, souvent conformes à des certifications de sécurité comme FIPS 140-2, et ils sont conçus pour fonctionner dans des environnements serveurs ou des centres de données.

Un HSM se distingue donc d'un *hardware wallet* par son usage : il est conçu pour des environnements serveur et des opérations automatisées à haute fréquence, alors qu'un *hardware wallet* est destiné à un usage individuel et interactif. Cette différence d'usage implique également une différence de déploiement : le HSM s'intègre dans une infrastructure réseau, tandis que le *hardware wallet* constitue un périphérique personnel connecté ponctuellement.

Terme associé : **HARDWARE WALLET**

HTLC

LIGHTNING NETWORK

Sigle de « *Hashed Timelock Contract* ». C'est un mécanisme de contrat intelligent principalement utilisé sur Lightning. On le retrouve aussi parfois dans des *atomic swaps*. Fondamentalement, le HTLC permet de conditionner un transfert de fonds à la révélation d'un secret, et inclut également une condition temporelle pour protéger l'argent de l'envoyeur en cas d'échec de la transaction.

Sur Lightning, les HTLC permettent d'envoyer des bitcoins à un nœud avec lequel on ne dispose pas de canal direct, en passant à travers plusieurs canaux, sans besoin de confiance tout au long de la route. Le paiement entre chaque nœud est conditionné par la fourniture d'une préimage (le secret qui, lorsqu'il est haché, correspond à une valeur convenue). Si le destinataire final fournit cette préimage, il peut réclamer les fonds, et permet forcément à chaque nœud intermédiaire de réclamer, eux aussi, les fonds en cascade.

Par exemple, si Alice veut envoyer 1 BTC à David, mais qu'elle n'a pas de canal direct avec lui, elle doit passer par Bob et Carol, qui ont des canaux de paiement entre eux. Voici comment la transaction se déroule : *David présente une* invoice* Lightning à Alice. Celle-ci contient le hachage h d'un secret s (la préimage) que seul David connait. On a donc : $h = hachage(s)$;

- Alice crée un HTLC avec Bob, qui propose de lui envoyer 1 BTC à condition que Bob lui fournisse un secret s (la préimage) qui correspond au hachage h ;
- Bob crée un HTLC avec Carol, qui propose de lui envoyer 1 BTC à condition qu'elle fournisse le même secret s ;
- Carol crée un HTLC avec David, qui propose encore 1 BTC s'il révèle la préimage s ;
- David révèle s (qu'il était le seul à connaître) à Carol pour recevoir 1 BTC. Carol peut désormais utiliser s pour récupérer le BTC auprès de Bob. Et Bob, qui connait maintenant s, fait de même avec Alice.

Finalement, Alice a envoyé 1 BTC à David en passant par Bob et Carol (une action neutre pour ces derniers), sans que personne ait à se faire confiance, car tout est sécurisé en cascade par les conditions des HTLCs.

Les HTLCs permettent ainsi de faire des échanges dits « atomiques » : soit le transfert est entièrement réussi, soit il échoue, et les fonds sont restitués. Cela garantit la sécurité des transactions même entre plusieurs participants sans besoin de confiance.

Termes associés : LIGHTNING NETWORK | TIMELOCK

HTLC ENDORSEMENT

LIGHTNING NETWORK | FR : ENDOSSEMENT DE HTLC

Mécanisme de réputation locale sur le Lightning Network conçu pour atténuer les attaques de blocage de canal (*channel jamming*). Son principe consiste à laisser chaque noeud de routage marquer un HTLC entrant comme « endossé » ou non, selon l'historique de rentabilité du pair qui le transmet.

Lorsqu'un noeud reçoit un HTLC d'un pair qui s'est montré fiable et rentable par le passé, il peut transmettre cet endossement au saut suivant. Les HTLC endossés bénéficient alors d'un accès prioritaire à deux ressources limitées de chaque canal : les *slots* de HTLC (nombre maximal de paiements en attente) et la liquidité disponible. Une part de ces ressources est réservée exclusivement aux HTLC endossés, tandis que le reste demeure accessible à tous.

Ce mécanisme crée une dissuasion économique contre les attaquants. Un pair qui mène une attaque de *jamming* réduit sa rentabilité perçue par ses voisins, ce qui empêche ses futurs HTLC d'obtenir l'endossement et donc l'accès aux ressources protégées. Les noeuds honnêtes, dont les paiements se résolvent normalement, conservent en revanche leur statut d'endossement. L'approche est purement locale : chaque noeud évalue indépendamment ses pairs, sans registre global de réputation.

Termes associés : **CANAL DE PAIMENT** | **GOSSIP** | **LIGHTNING NETWORK**

HUMAN RIGHTS FOUNDATION

ORGANISATION | FR : FONDATION DES DROITS DE L'HOMME

Organisation non partisane à but non lucratif fondée en 2005 par Thor Halvorssen, dédiée à la promotion des droits de l'homme dans le monde, avec un accent particulier sur les sociétés autoritaires. Depuis 2020, la fondation gère un *Bitcoin Development Fund* qui finance des projets open source visant à améliorer la confidentialité, la décentralisation et l'accessibilité de Bitcoin. La HRF considère Bitcoin comme un outil d'émancipation financière pour les populations vivant sous des régimes autoritaires, où l'accès au système bancaire traditionnel est restreint ou surveillé. La fondation organise également des événements éducatifs autour de Bitcoin et des libertés individuelles.

HWI

PORTEFEUILLE

Sigle de « *Hardware Wallet Interface* ». C'est une interface standardisée permettant l'intégration et l'interaction entre des logiciels de gestion de portefeuilles Bitcoin et des portefeuilles matériels (hardware wallets). Plus précisément, HWI est à la fois une bibliothèque en Python et un outil en ligne de commande. Il facilite la communication entre ces composants en utilisant des PSBTs (transactions Bitcoin partiellement signées) et éventuellement des Descriptors (output script descriptors). Cette interface a d'abord été développée pour Bitcoin Core, puis elle est devenue un standard utilisé par la plupart des logiciels de portefeuilles.

Termes associés : **HARDWARE WALLET** | **OUTPUT SCRIPT DESCRIPTORS** | **PSBT**

HYDRO COOLING

↗ MINAGE | FR : REFROIDISSEMENT À L'EAU

Système de refroidissement pour les ASICs qui utilise un circuit fermé où de l'eau circule à travers des blocs de refroidissement, des tuyaux, puis un radiateur. L'objectif est d'extraire la chaleur des composants et de maintenir une bonne température pour leur fonctionnement.

Termes associés : **ASIC** | **IMMERSION COOLING**

HYPER-BITCOINISATION

Ⓢ ÉCONOMIE ET RÉGULATION | EN : HYPERBITCOINIZATION

Concept théorique décrivant un processus d'adoption massive et irréversible de Bitcoin comme monnaie dominante au niveau mondial, remplaçant progressivement les monnaies fiat. Le terme a été popularisé par Daniel Krawisz en 2014, qui le décrivait comme une transition volontaire d'une monnaie inférieure vers une monnaie supérieure, ou plus précisément comme une « démonétisation induite par Bitcoin » (*Bitcoin-induced currency demonetization*).

L'hyper-bitcoinisation suppose un scénario dans lequel la perte de confiance dans les monnaies fiat (due à l'inflation, au contrôle financier ou à l'instabilité monétaire) pousse les individus et les institutions à adopter Bitcoin massivement. L'hyper-bitcoinisation serait un mouvement volontaire et décentralisé, dans lequel chaque individu choisit librement de migrer vers un standard monétaire plus solide.

Ce concept est débattu au sein même de la communauté Bitcoin : certains le considèrent comme un objectif inévitable à long terme, tandis que d'autres le voient comme un idéal théorique peu probable dans sa forme absolue, Bitcoin pouvant coexister avec d'autres systèmes monétaires, ou bien simplement rester une monnaie utilisée à la marge, uniquement dans la contre-économie.

Termes associés : **FIAT** | **HYPERINFLATION**

HYPERINFLATION

Ⓢ ÉCONOMIE ET RÉGULATION

Phénomène économique caractérisé par une hausse des prix extrêmement rapide et incontrôlable. L'hyperinflation survient lorsqu'un gouvernement ou une banque centrale émet de la monnaie en quantités excessives, souvent pour financer des déficits budgétaires, entraînant une perte de confiance totale dans la devise nationale.

Les exemples historiques les plus marquants sont la République de Weimar en Allemagne (1923), le Zimbabwe dans les années 2000, ou encore le Venezuela à partir de 2016. Dans ces cas, la monnaie a perdu sa fonction de réserve de valeur et de moyen d'échange, poussant les populations vers des monnaies alternatives ou le troc.

Le risque d'hyperinflation est un argument central des bitcoiners, qui mettent en avant l'offre limitée à 21 millions d'unités de BTC comme protection contre ce type de dérives monétaires. Contrairement aux monnaies fiat, aucune autorité

ne peut augmenter arbitrairement la quantité de bitcoins en circulation.

Termes associés : **FIAT** **INFLATION**

I

I2P

RÉSEAU

Réseau de communication anonyme conçu pour assurer la confidentialité des échanges sur Internet. Comme Tor, I2P chiffre les données du réseau et utilise un système de routage appelé *garlic routing*. Cette méthode garantit l'anonymat des communications en les faisant passer par des serveurs intermédiaires qui redirigent les informations sans connaître l'identité de l'émetteur ni celle du destinataire. Tout comme Tor, I2P peut être utilisé sur un nœud Bitcoin, afin de sécuriser ses communications réseau.

Le sigle « I2P » signifie « Invisible Internet Project ».

Termes associés : **NOEUD** **TOR**

IBD - INITIAL BLOCK DOWNLOAD

PROTOCOLE | FR : SYNCHRONISATION INITIALE

Fait référence au processus par lequel un nœud télécharge et vérifie la blockchain depuis le bloc de Genèse, et se synchronise aux autres nœuds du réseau Bitcoin. L'IBD doit être réalisée au lancement d'un nouveau nœud complet. Au début de cette synchronisation initiale, le nœud ne dispose d'aucune information sur les transactions précédentes. Il télécharge donc chaque bloc depuis les autres nœuds du réseau, vérifie sa validité, puis l'ajoute à sa version locale de la blockchain. Il convient de noter que l'IBD peut être longue et exigeante en ressources en raison de la taille croissante de la blockchain et de l'UTXO set. La rapidité de son exécution dépend des capacités de calcul de l'ordinateur qui héberge le nœud, de ses capacités en RAM, de la vitesse du dispositif de stockage et de la bande passante. Pour vous donner une idée, si vous disposez d'une connexion internet puissante, et que le nœud est hébergé sur le dernier MacBook avec beaucoup de RAM, l'IBD ne prendra que quelques heures. En revanche, si vous utilisez un micro-ordinateur comme un Raspberry Pi, l'IBD pourra prendre une semaine ou plus.

En français, il est globalement admis de parler directement d'un(e) IBD.

Termes associés : **NOEUD COMPLET** **UTXO SET**

IMMERSION COOLING

MINAGE | FR : REFROIDISSEMENT PAR IMMERSION

Technique de refroidissement utilisée dans le cadre du minage, où les ASICs sont immergés dans un liquide non-conducteur pour dissiper la chaleur qu'ils produisent. L'immersion présente plusieurs avantages par rapport au refroidissement à l'air. Tout d'abord, il n'y a pas besoin d'utiliser des ventilateurs, ce qui réduit ainsi la consommation énergétique. Aussi, les machines sont protégées de la poussière, ce qui améliore leur performance et leur longévité. L'immersion permet également une meilleure dissipation de la chaleur, car les composants sont directement en contact avec le fluide de refroidissement.

Termes associés : **AIR COOLING** **ASIC**

IMPLÉMENTATION DE BITCOIN

PROTOCOLE | EN : BITCOIN IMPLEMENTATION

Fait référence à un logiciel qui applique et suit les règles définies par le protocole Bitcoin. Ce que l'on appelle « Bitcoin », c'est généralement le système d'argent électronique. C'est un protocole qui spécifie des règles. Il est représenté, concrètement, par des nœuds qui forment un réseau. Ce système ne dispose pas spécifiquement de code. C'est simplement un ensemble de grandes règles tacites imposées par le consensus des utilisateurs via leurs nœuds. En revanche, ceux qui disposent de code informatique, et qui peuvent donc être développés, maintenus et modifiés, ce sont plutôt les logiciels de nœuds Bitcoin. Ce sont des implémentations indépendantes du protocole Bitcoin, qui peuvent se connecter au reste du réseau.

Parmi les implémentations de Bitcoin, Bitcoin Core est de loin la plus répandue, puisqu'elle représente la très grande majorité du réseau de nœuds. Cependant, il existe aussi des implémentations minoritaires telles que btcd, Bcoin, et Bitcoin Knots. Malgré la diversité des clients logiciels disponibles, la prédominance de Bitcoin Core sur le réseau lui confère une influence presque exclusive sur l'application du protocole Bitcoin. On dit donc que Bitcoin Core représente dans les faits le protocole Bitcoin lui-même.

Termes associés : **BITCOIN CORE** | **BITCOIN KNOTS**

INBOUND CAPACITY

LIGHTNING NETWORK | FR : CAPACITÉ ENTRANTE

Désigne la quantité maximale de bitcoins qu'un nœud peut recevoir à travers un canal spécifique sur le Lightning Network. Elle dépend des fonds que le nœud pair a engagés dans le canal lors de son ouverture, ou que l'on a envoyé lors d'un paiement Lightning sortant ou d'un routage.

Termes associés : **LIGHTNING NETWORK** | **OUTBOUND CAPACITY**

INDEX - KEY

PORTEFEUILLE | FR : NUMÉRO DE CLÉ

Dans le contexte d'un portefeuille HD, fait référence au numéro séquentiel attribué à une clé enfant générée à partir d'une clé parent. Cet index est utilisé en combinaison avec la clé parent et le code chaîne parent pour dériver de manière déterministe des clés enfants uniques. Il permet une organisation structurée et la génération reproductible de multiples paires de clés enfants sœurs depuis une même clé parent. C'est un entier de 32 bits utilisé dans la fonction de dérivation `HMAC-SHA512`. Ce nombre permet donc de différencier les clés enfants sœurs au sein d'un portefeuille HD.

Termes associés : **CHEMIN DE DÉRIVATION** | **DÉRIVATION**

INDEXES/TXINDEX/

PROTOCOLE

Fichiers dans Bitcoin Core qui sont dédiés à l'indexation de toutes les transactions présentes dans la blockchain. Cette indexation permet de rechercher rapidement des informations détaillées sur une transaction en utilisant son identifiant (TXID), sans avoir à parcourir l'intégralité de la blockchain. La création de ces fichiers d'indexation est une option non activée par défaut dans Bitcoin Core. Si cette fonctionnalité n'est pas activée, votre nœud indexera uniquement les transactions associées aux portefeuilles rattachés à votre nœud. Pour activer l'indexation de toutes les transactions, il faut régler le paramètre `txindex=1` dans le fichier `bitcoin.conf`. Cette option est particulièrement utile pour les applications et les services qui font des recherches fréquentes dans l'historique des transactions de Bitcoin.

Termes associés : **BITCOIN CORE** | **BITCOIN.CONF**

INFLATION

ÉCONOMIE ET RÉGULATION

Désigne l'augmentation de la masse monétaire en circulation, entraînant une baisse de la valeur de chaque unité de monnaie et une hausse des prix des biens et services. Pour les monnaies étatiques, ce phénomène survient lorsque les banques centrales émettent de nouvelles unités monétaires.

Dans le contexte de Bitcoin, l'inflation est contrôlée par le protocole. De nouveaux bitcoins sont générés à chaque validation de bloc via la subvention attribuée au mineur victorieux. Cette subvention était initialement de 50 BTC par bloc et elle est réduite de moitié tous les 210 000 blocs lors d'un événement connu sous le nom de « halving ». L'émission totale de bitcoins est plafonnée à environ 21 millions d'unités, ce qui signifie que l'inflation monétaire du BTC est décroissante et s'arrêtera une fois que le nombre maximal d'unités sera atteint.

Termes associés : **HALVING** | **SUBVENTION DE BLOC**

INPUT

PROTOCOLE | FR : ENTRÉE

Dans le contexte de Bitcoin, un input (entrée) au sein d'une transaction fait référence aux UTXOs (*Unspent Transaction Outputs*) utilisés comme fonds d'origine pour satisfaire les sorties (outputs). Chaque input contient des références aux UTXOs précédents, qui seront alors consommés par la transaction. Ces inputs sont utilisés pour alimenter de nouveaux UTXOs qui seront créés comme outputs de la transaction, et qui peuvent ensuite être dépensés dans des transactions futures.

Le rôle de la transaction Bitcoin est donc de consommer des UTXOs en entrées, et de créer des nouveaux UTXO en sorties. La différence entre les deux correspond aux frais de transactions qui peuvent être récupérés par le mineur qui valide le bloc.

D'un point de vue plus large, en informatique, le terme « input » ou « entrée » désigne généralement les données fournies à une fonction, un algorithme, ou

un système en tant qu'opérandes ou informations requises pour effectuer une opération ou un calcul. Dans ce sens, le terme est utilisé de manière plus générique pour décrire tout ce qui est fourni à un processus en vue d'obtenir un résultat ou une sortie (output). Par exemple, lorsque l'on passe une donnée dans une fonction de hachage cryptographique, cette information est nommée « entrée » ou « input ».

Termes associés : **OUTPUT** **UTXO**

INSCRIPTIONS

COUCHE SUPÉRIEURE

Dans le cadre de la théorie des Ordinals, les inscriptions sont des contenus arbitraires gravés sur des sats, transformant ces derniers en artefacts numériques natifs de Bitcoin. Les inscriptions sont effectuées via des transactions qui exposent le contenu de l'information dans le script d'un input Taproot de cette manière :

```
OP_FALSE
OP_IF
OP_PUSH "ord"
OP_PUSH 1
OP_PUSH "<content type>"
OP_PUSH 0
OP_PUSH "<la data ici>"
OP_ENDIF
```

Ces artefacts numériques, comme des NFTs, peuvent être échangés et conservés.

Termes associés : **DIGITAL ARTIFACTS** **ORDINALS THEORY**

INTERFACE

RGB

Dans le cadre du protocole RGB, l'*Interface* désigne l'ensemble des instructions qui décodent les données binaires compilées dans un *Schema* ou au sein des opérations et états d'un contrat. Elle sert de couche d'interprétation, afin de transformer les informations brutes en une représentation lisible par l'utilisateur ou son *wallet*.

Termes associés : **INTERFACE IMPLEMENTATION** **SCHEMA**

INTERFACE IMPLEMENTATION

RGB

Dans le cadre du protocole RGB, l'*Interface Implementation* désigne l'ensemble des déclarations établissant la correspondance entre une *Interface* et un *Schema*. Elle permet de réaliser la traduction sémantique des données binaires d'un contrat, afin d'assurer leur interprétation correcte par l'utilisateur ou par les applications, telles que les *wallets*.

Termes associés : **INTERFACE** **SCHEMA**

INVOICE LIGHTNING

LIGHTNING NETWORK | FR : FACTURE

Requête de paiement Lightning générée par le destinataire, qui contient toutes les informations nécessaires pour réaliser la transaction.

Une invoice Lightning contient la destination du paiement sous la forme de la clé publique du nœud destinataire, mais également un préfixe `ln`, le montant, un temps avant expiration, le hachage du secret utilisé dans le cadre des HTLCs, ainsi que d'autres métadonnées, pour certaines optionnelles, comme des options relatives au routage. Ces invoices sont définies par la norme BOLT-11. Une fois payée, une invoice Lightning ne peut plus être réutilisée.

Termes associés : **BOLT** **HTLC**

INVOICE RGB

RGB

Dans le cadre du protocole RGB, une *Invoice* se présente sous la forme d'une URI préfixée par `rgb:`, qui intègre les informations indispensables à la construction d'une *State Transition* par le payeur. En pratique, il s'agit d'une facture qui permet à la contrepartie de générer la transition adéquate pour transférer un actif ou actualiser l'état d'un contrat.

Termes associés : **CONSIGNMENT** **STATE TRANSITION**

IOU

ÉCONOMIE ET RÉGULATION

Sigle de l'anglais *I Owe You* (« Je te dois ») utilisé dans le contexte de Bitcoin pour désigner des actifs numériques qui sont adossés à des actifs sous-jacents auxquels ils devraient normalement être indexés. Ce terme s'applique notamment aux *stablecoins* ou aux représentations de BTC sur des systèmes externes à Bitcoin, tels que les *sidechains*, les *drivechains*, les plateformes d'échange, ou encore les ETF. Ces actifs numériques représentent une promesse de valeur équivalente à celle de l'actif sous-jacent.

Termes associés : **SIDECHAIN** **STABLECOIN**

IP_ASN.MAP

RÉSEAU

Fichier utilisé dans Bitcoin Core pour stocker l'ASMAP qui permet d'améliorer le *bucketing* (c'est-à-dire, le regroupement) des adresses IP, en se basant sur les numéros de systèmes autonomes (ASN). Plutôt que de regrouper les connexions sortantes par préfixes de réseau IP (`/16`), ce fichier permet de diversifier les connexions en établissant une carte d'adressage IP vers les ASN, qui sont des identifiants uniques pour chaque réseau sur Internet. L'idée est d'améliorer la sécurité et la topologie du réseau Bitcoin en diversifiant les connexions pour se prémunir contre certaines attaques (notamment l'attaque Erebus).

Termes associés : **ASMAP** **EREBUS**

IP REPORT

↗ MINAGE

Bouton présent sur les ASICs qui permet de trouver rapidement l'adresse IP de la machine sur le réseau local. Pour utiliser cette fonctionnalité, il faut appuyer sur le bouton *IP Report* de l'ASIC pendant quelques secondes. Ensuite, en utilisant un logiciel spécialisé sur un ordinateur connecté au même réseau, l'adresse IP du mineur sera affichée à l'écran. Ce bouton permet notamment de localiser facilement une machine sur le réseau local lorsque l'on a plusieurs machines.

Termes associés : **ASIC** **MINEUR**

ISSUE

</> INFORMATIQUE

Dans le cadre de GitHub et d'autres plateformes d'hébergement de code, une issue est un rapport qui signale un bug, propose une amélioration ou suggère une nouvelle fonctionnalité. Elle sert de point de discussion pour les contributeurs et permet de suivre les tâches à accomplir ou les problèmes à résoudre dans le projet. En tant qu'utilisateur de Bitcoin, vous pouvez aider les logiciels open source que vous utilisez en signalant d'éventuels bugs via des issues sur le dépôt du projet.

Termes associés : **GITHUB** **PULL REQUEST**

J

JADE

PORTEFEUILLE

Hardware wallet fabriqué par Blockstream, conçu pour sécuriser des clés privées Bitcoin et compatible avec le réseau Liquid. Le Jade se distingue par un modèle de sécurité original : au lieu d'utiliser une puce *secure element*, il repose sur un modèle de « *virtual secure element* » où la clé privée est chiffrée et ne peut être déchiffrée qu'en combinant un secret stocké sur l'appareil avec un secret détenu par le serveur de Blockstream (ou un serveur personnel).

Les firmwares Jade sont entièrement open source. Les appareils disposent d'une caméra pour le mode *air-gapped* via QR codes, et fonctionnent avec Blockstream Green, Sparrow Wallet ou d'autres logiciels compatibles. Il existe 2 hardware wallets dans la famille Jade : le Blockstream Jade classique, disponible prémonté ou en DIY, et le Blockstream Jade Plus, le modèle le plus récent et plus premium, doté d'un grand écran couleur et d'une UX soignée.

Termes associés : **HARDWARE WALLET** | **PORTEFEUILLE**

JAM

CONFIDENTIALITÉ

JAM est une interface web pour JoinMarket axée sur la facilité d'utilisation. JoinMarket est une implémentation de coinjoin sur Bitcoin, mais son utilisation en ligne de commandes peut effrayer les utilisateurs débutants. JAM a justement été créé pour rendre les fonctionnalités avancées de JoinMarket accessibles aux débutants.

Termes associés : **COINJOIN** | **JOININBOX** | **JOINMARKET**

JAN3

ORGANISATION

Entreprise fondée en 2022 par Samson Mow, ancien directeur stratégique de Blockstream. Le nom « JAN3 » fait référence au 3 janvier 2009, date à laquelle Satoshi Nakamoto a créé le bloc genesis de Bitcoin. Jan3 se consacre à l'accélération de l'adoption de Bitcoin au niveau des États-nations, en développant des infrastructures basées sur des surcouches comme le Lightning Network et des sidechains comme le réseau Liquid. L'entreprise collabore notamment avec le Salvador et la région autonome de Madère pour l'intégration de Bitcoin dans leurs systèmes financiers. Jan3 est également actif dans le développement logiciel, notamment avec le portefeuille Aqua.

Termes associés : **LIGHTNING NETWORK** | **LIQUID NETWORK**

JBOK

PORTEFEUILLE

Les portefeuilles JBOK, acronyme pour « *Just a Bunch Of Keys* » (en français « juste un trousseau de clés »), font référence aux vieux portefeuilles Bitcoin qui stockaient un ensemble de paires de clés générées de manière indépendante et pseudo-aléatoire. Contrairement aux portefeuilles HD modernes, qui génèrent des clés de manière déterministe et hiérarchique à partir d'une graine unique, les portefeuilles JBOK ne présentaient aucune relation hiérarchique ou déterministe entre les clés. Elles étaient toutes indépendantes les unes des autres. En raison de leur gestion moins efficace et de la difficulté de sauvegarde, ces portefeuilles sont devenus obsolètes et ont été spontanément remplacés par des solutions HD plus avancées, comme standardisées dans le BIP-0032.

Termes associés : **BIP-0032** | **GRAINE** | **HD - HIERARCHICAL-DETERMINISTIC**

JIT CHANNEL

LIGHTNING NETWORK | FR : CANAL JIT

Canal Lightning ouvert à la volée par un fournisseur de services Lightning (LSP) au moment où un paiement entrant arrive pour un client qui ne dispose pas encore de canal. Le terme « JIT » (*just-in-time*) signifie que le canal n'est créé qu'au moment précis où il est nécessaire, et non en avance.

Le fonctionnement repose sur le protocole LSPS2 (bLIP-0052). Le client demande au LSP les paramètres de frais d'ouverture, puis obtient un identifiant SCID temporaire qu'il inclut dans son invoice Lightning comme indice de routage. Lorsqu'un payeur envoie le paiement, le LSP intercepte le HTLC, ouvre un canal à zéro confirmation vers le client et transfère le paiement en déduisant les frais d'ouverture du montant reçu. Le client réclame le paiement en révélant la préimage.

Deux modèles de confiance existent :

- dans le modèle par défaut, le LSP diffuse la transaction de financement avant de recevoir la préimage, ce qui laisse la possibilité au client de vérifier sa validité ;
- dans le modèle alternatif, le client transmet la préimage immédiatement après l'engagement du HTLC.

Ce mécanisme ne doit pas être confondu avec le *JIT routing*, qui consiste à rééquilibrer des canaux existants.

Termes associés : **BLIP-0052** | **TRANSACTION DE FINANCEMENT**

JIT ROUTING

LIGHTNING NETWORK | FR : ROUTAGE JIT

Technique proposée par René Pickhardt en 2019 qui permet à un nœud de routage Lightning de rééquilibrer ses canaux à la volée pour transmettre un paiement qui aurait autrement été refusé faute de liquidité suffisante sur le canal sortant. Le terme « JIT » (*just-in-time*) indique que le rééquilibrage est déclenché au moment précis où le HTLC entrant arrive, et non de manière préventive.

Concrètement, lorsqu'un nœud intermédiaire reçoit un HTLC à transmettre mais manque de fonds sur le canal de sortie, il suspend brièvement le relais et effectue un paiement circulaire à travers ses autres canaux pour déplacer la liquidité nécessaire. Le nœud exploite sa connaissance locale du réseau pour calculer rapidement des cycles de rééquilibrage peu coûteux. Si le rééquilibrage réussit et que les frais engagés sont inférieurs aux frais de routage perçus, l'opération est rentable.

Le *JIT routing* ne nécessite aucune modification de protocole : il s'agit d'un comportement local que chaque nœud de routage peut adopter indépendamment.

Ce mécanisme ne doit pas être confondu avec les *JIT channels*, qui désignent l'ouverture à la volée d'un canal par un LSP.

Termes associés : **JIT CHANNEL** | **ROUTAGE EN OIGNON**

JOININBOX

CONFIDENTIALITÉ

Environnement Linux minimaliste conçu pour utiliser JoinMarket. JoininBox dispose d'un menu établi sur le terminal, et dispose de toutes les commandes de base de JoinMarket. Il permet d'utiliser l'implémentation de coinjoin JoinMarket sur une machine dédiée à cette tâche, comme un Raspberry Pi par exemple.

Termes associés : **COINJOIN** | **JAM** | **JOINMARKET**

JOINMARKET

CONFIDENTIALITÉ

Implémentation de coinjoin créée en 2015 par Chris Belcher. JoinMarket se distingue des autres implémentations de coinjoin par son modèle unique de mise en relation des utilisateurs. Il fonctionne comme un marché P2P où les « *makers* » mettent leurs bitcoins à disposition pour le mixage et reçoivent des frais, tandis que les « *takers* » utilisent ces liquidités moyennant une rémunération pour réaliser des coinjoins avec.

Terme associé : **CHAUMIAN COINJOIN**

JOINPOOL

COUCHE SUPÉRIEURE

Type de système de paiement en surcouche basé sur le partage de la propriété d'un ou plusieurs UTXO, sans besoin de confiance. Lorsque des fonds sont sortis d'une pool dans ce type de système, il doit être impossible de déterminer on-chain quel utilisateur a effectué la dépense.

Le principe fondamental d'un système de type joinpool est donc le partage d'UTXO pour améliorer le passage à l'échelle de Bitcoin : plutôt que chaque participant possède un ou plusieurs UTXO individuels, un seul UTXO représente les fonds de l'ensemble du groupe. Chaque membre conserve la capacité de retirer unilatéralement ses fonds à tout moment, soit via des transactions présignées, soit grâce à des mécanismes proposés au niveau du protocole comme les covenants. Sans covenants, l'interactivité requise entre participants croît significativement, car toute modification de l'état du pool nécessite la signature de tous les membres.

Le protocole Ark peut être considéré comme une implémentation concrète du concept de joinpool, où un *Ark Service Provider* joue le rôle de coordinateur et fournisseur de liquidité pour les participants du pool.

Ce type de système est parfois également désigné sous les noms de « *coinpool* » ou « *payment pool* ».

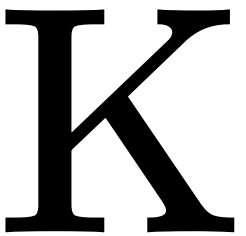
Terme associé : **ARK**

JOINSTR

CONFIDENTIALITÉ

Protocole de coinjoin décentralisé qui utilise Nostr comme couche de communication. Proposé en août 2022 par le développeur pseudonyme « 1440000bytes », il élimine le besoin d'un coordinateur central, qui constitue traditionnellement un point de défaillance dans les implémentations de coinjoin. Les participants inscrivent leurs sorties via des relais Nostr, puis le système crée une transaction Bitcoin partiellement signée (PSBT) que chaque participant signe individuellement. Une fois toutes les signatures collectées, la transaction combinée est diffusée sur le réseau.

Termes associés : **COINJOIN** **NOSTR**



KEY TELEPORT

PORTEFEUILLE

Fonctionnalité des Coldcard qui permet le transfert chiffré de données confidentielles entre deux appareils. Les données transférables comprennent les *seeds* (*seed* principale ou *Seed Vault*), les notes et mots de passe, les sauvegardes complètes et les PSBT multisig.

Termes associés : **BBQR** | **COLDCARD**

KEYNÉSIANISME

ÉCONOMIE ET RÉGULATION | EN : KEYNESIANISM

Courant de pensée économique fondé sur les travaux de l'économiste britannique John Maynard Keynes. Le keynésianisme préconise l'intervention active de l'État dans l'économie pour stabiliser les cycles économiques. Les politiques keynésiennes reposent sur la capacité des banques centrales à ajuster la masse monétaire et les taux d'intérêt, ainsi que sur la capacité des gouvernements à emprunter et dépenser. Ce cadre théorique domine largement la politique économique contemporaine.

Dans la communauté Bitcoin, le keynésianisme est souvent critiqué. Les bitcoiners, généralement proches de l'école autrichienne, lui reprochent d'encourager l'expansion monétaire, de fausser les signaux de prix par la manipulation des taux d'intérêt, de favoriser les mal-investissements et de générer une dette publique croissante. Bitcoin, avec son offre fixe et l'impossibilité de le manipuler monétairement, est fondamentalement incompatible avec les outils de politique monétaire keynésiens.

Terme associé : **ÉCOLE AUTRICHIENNE**

KEYSEND

LIGHTNING NETWORK | FR : PAIEMENT SPONTANÉ

Méthode de paiement sur le Lightning Network permettant d'envoyer des fonds directement vers la clé publique d'un nœud, sans que le destinataire ait besoin de générer une facture au préalable. Contrairement au flux classique BOLT-11 où le destinataire choisit la préimage et la transmet via la facture, c'est ici l'expéditeur qui génère la préimage et l'inclut dans le paquet onion chiffré du paiement. Seul le nœud destinataire peut déchiffrer le paquet et récupérer la préimage pour finaliser le HTLC.

Le nœud destinataire doit activer explicitement la réception de keysend (`accept-keysend=true` dans la configuration LND). Des données arbitraires peuvent être jointes au paiement via des enregistrements TLV, ce qui rend keysend utile pour les dons, les paiements en streaming (Podcasting 2.0) ou la messagerie entre nœuds. La principale limite est l'absence de preuve de paiement : puisque l'expéditeur choisit lui-même la préimage, il ne peut pas prouver après coup qu'il a effectué le paiement.

Terme associé : **ROUTAGE EN OIGNON**

KEYSTORE

PORTEFEUILLE

Dans le contexte d'un logiciel de portefeuille Bitcoin, un keystore est une structure qui regroupe les informations nécessaires au suivi et à l'utilisation d'un jeu de clés. Il contient généralement un label, une empreinte de clé maîtresse (*master fingerprint*), un chemin de dérivation et une clé publique étendue (`xpub`). Ces éléments permettent au portefeuille de générer les adresses de réception et de surveiller les transactions qui leur sont associées.

Termes associés : **MULTISIG** | **PORTEFEUILLE** | **XPUB**

KNAPSACK SOLVER

PORTEFEUILLE

Ancienne méthode utilisée pour la sélection de pièces dans le portefeuille de Bitcoin Core avant la version 0.17. Le *Knapsack Solver* tente de résoudre le problème de sélection de pièces en choisissant de manière itérative et aléatoire des UTXOs, et en les additionnant par sous-ensembles, dans l'objectif de minimiser les frais et la taille de la transaction. Cette méthode a depuis été remplacée par le *Branch-and-Bound*.

Terme associé : **BRANCH-AND-BOUND**

KRUX

OUTIL

Firmware open source qui transforme des appareils à microcontrôleur Kendryte K210, comme le Maix Amigo ou le M5StickV, en dispositifs de signature de transactions Bitcoin. Krux permet de générer et de gérer des clés privées, de signer des transactions hors ligne et d'interagir avec des portefeuilles coordinateurs (Sparrow, Specter...) via des QR codes ou des cartes microSD. Le projet s'adresse aux utilisateurs souhaitant construire un *hardware wallet* à faible coût à partir de matériel générique du commerce, tout en conservant un contrôle total sur le code source.

Termes associés : **BIP-0039** | **HARDWARE WALLET**

KYC - KNOW YOUR CUSTOMER

ÉCONOMIE ET RÉGULATION

Procédure réglementaire utilisée par certaines entreprises opérant sur Bitcoin pour vérifier l'identité de leurs clients dans le cadre de la lutte contre le blanchiment d'argent et le financement du terrorisme. Le KYC implique la collecte et la vérification de données personnelles. Dans le cadre de l'achat de bitcoins, le KYC engendre plusieurs risques pour l'utilisateur, notamment :

- Le risque de fuite de données personnelles en lien avec une activité sur Bitcoin. Le stockage d'informations sur les serveurs d'entreprises peut entraîner des fuites, exposant les données des utilisateurs à des tentatives d'hameçonnage, des attaques physiques ou une usurpation d'identité, notamment en raison de leur association avec l'environnement de Bitcoin ;
- L'exposition à la surveillance étatique. L'achat de BTC via des acteurs régulés peut révéler à l'État que l'utilisateur a possédé du bitcoin à un moment

donné, ce qui pourrait avoir des conséquences futures en cas de bouleversements politiques ou économiques ;

- La facilitation du traçage on-chain. La réalisation d'un KYC crée un lien direct entre l'identité de l'utilisateur et ses transactions sur la blockchain, permettant d'établir un point d'entrée pour une analyse de chaîne.

Termes associés : **AML - ANTI-MONEY LAUNDERING** | **ANALYSE DE CHAINE**

L

L402

LIGHTNING NETWORK

L402, anciennement connu sous le nom de LSAT (*Lightning Service Authentication Token*), est un standard d'authentification et de paiement pour les services distribués. Il tire son nom du code de statut HTTP 402 (*Payment Required*). L402 combine deux mécanismes : les macaroons, qui sont des jetons d'authentification porteur vérifiables par simple cryptographie sans base de données centralisée, et le Lightning Network, qui fournit un système de paiement instantané.

Le fonctionnement de L402 repose sur l'association d'un macaroon avec le hash d'un paiement Lightning. Lorsqu'un utilisateur souhaite accéder à un service protégé, celui-ci lui remet un macaroon accompagné d'une invoice Lightning. Une fois le paiement effectué, l'utilisateur obtient la préimage qui, combinée au macaroon, forme un identifiant L402 valide. Le service n'a pas besoin de vérifier directement si l'invoice a été payée : la simple présentation d'une préimage correspondant au hash du paiement suffit à prouver le règlement.

L402 est implémenté dans Aperture, un reverse proxy développé par Lightning Labs, qui gère l'émission des macaroons et des factures, puis transmet les requêtes authentifiées vers les API protégées.

Termes associés : **INVOICE LIGHTNING** | **LIGHTNING NETWORK** | **MACAROON**

LABEL

PORTEFEUILLE | FR : ÉTIQUETTE

Étiquette ou annotation attribuée à un UTXO dans un portefeuille Bitcoin pour se souvenir de sa provenance. Par exemple, si je possède un UTXO provenant d'un achat P2P sur Bisq avec Charles, je pourrais lui attribuer le label « `Non-KYC Bisq Charles` ». C'est une bonne pratique qui aide à se rappeler l'origine ou la destination prévue de cet UTXO, ce qui facilite la gestion des fonds et l'optimisation de la confidentialité. Le *labelling* est d'autant plus important lorsqu'il est utilisé avec le *coin control*. En effet, en permettant aux utilisateurs de différencier et de sélectionner précisément les UTXOs pour leurs transactions, cette pratique aide à éviter la fusion d'UTXOs provenant de sources différentes. Cela limite les risques associés à l'heuristique d'analyse de chaîne CIOH (*Common Input Ownership Heuristic*), qui peut révéler la propriété commune des entrées d'une transaction.

Terme associé : **COIN CONTROL**

LABEL - SILENT PAYMENTS

CONFIDENTIALITÉ	FR : ÉTIQUETTE - SILENT PAYMENTS

Dans le cadre du protocole Silent Payments, les labels sont des entiers utilisés pour modifier l'adresse statique initiale, afin de créer de nombreuses autres adresses statiques. L'utilisation de ces labels permet de ségréguer les paiements envoyés via Silent Payments, en employant des adresses statiques différentes pour chaque usage, sans augmenter excessivement la charge opérationnelle pour la détection de ces paiements (scanning). Bob utilise une adresse statique B, composée de deux clés publiques : B_{scan} pour le scan et B_{spend} pour la dépense. On ajoute le hachage de b_{scan} et d'un entier m, multipliés scalairement par le point générateur G, à la clé publique de dépense B_{spend} pour créer une sorte de nouvelle clé publique de dépense B_m :

$$B_m \ = \ B_{spend} \ + \ hash(b_{scan} \parallel m) \cdot G$$

Par exemple, la première clé B_1 est obtenue de cette manière :

$$B_1 \ = \ B_{spend} \ + \ hash(b_{scan} \parallel 1) \cdot G$$

L'adresse statique publiée par Bob sera dorénavant composée de B_{scan} et de B_m. Par exemple, la première adresse statique avec le label 1 sera :

$$B \ = \ B_{scan} \parallel B_1$$

On commence seulement à partir du label 1, car le label 0 est réservé pour le change. Alice, qui souhaite envoyer des bitcoins sur l'adresse statique labellisée transmise par Bob, va dériver l'adresse de paiement unique P_0 en utilisant la nouvelle B_1 à la place de B_{spend} :

$$P_0 \ = \ B_1 \ + \ hash(inputHash \cdot a \cdot B_{scan} \parallel 0) \cdot G$$

En réalité, Alice ne sait même pas forcément que Bob dispose d'une adresse labelisée, car elle utilise simplement la seconde partie de l'adresse statique qu'il lui a fourni, et en l'occurrence, c'est la valeur B_1 plutôt que B_{spend}. Pour scanner les paiements, Bob va toujours utiliser la valeur de son adresse statique initiale avec B_{spend} de cette manière :

$$P_0 \ = \ B_{spend} \ + \ hash(inputHash \cdot b_{scan} \cdot A \parallel 0) \cdot G$$

Puis, il va simplement soustraire la valeur qu'il trouve pour P_0 de chaque output un à un. Il vérifie ensuite si un des résultats de ces soustractions correspond à la valeur d'un des labels qu'il utilise sur son portefeuille. Si ça matche, par exemple, pour l'output #4 avec le label 1, cela veut dire que cet output est un Silent Payment associé à son adresse statique labelisée B_1 :

$$Out_4 \ - \ P_0 \ = \ hash(b_{scan} \parallel 1) \cdot G$$

Cela fonctionne, car :

$$B_1 = B_{spend} + hash(b_{scan} \parallel 1) \cdot G$$

Grâce à cette méthode, Bob peut utiliser une multitude d'adresses statiques (avec B_1, B_2, B_3...), toutes dérivées depuis son adresse statique de base ($B = B_{scan} \parallel B_{spend}$), afin de bien séparer les usages.

Attention toutefois, cette séparation des adresses statiques vaut uniquement d'un point de vue de gestion personnelle du portefeuille, mais ne permet pas de séparer les identités. Puisqu'elles disposent toutes du même B_{scan}, il est très facile d'associer toutes les adresses statiques ensemble et de déduire qu'elles appartiennent à une unique entité.

Terme associé : **SILENT PAYMENTS**

LATENCE

RÉSEAU | EN : LATENCY

Délai entre l'émission et la réception d'une information. En informatique, elle représente le temps nécessaire pour que des données voyagent entre deux ordinateurs. Dans le contexte de Bitcoin, la latence peut être étudiée au niveau du délai de diffusion des blocs à travers le réseau. Une latence élevée peut entraîner des désavantages pour les mineurs isolés, car ceux-ci continuent de travailler sur des blocs déjà invalidés par de nouveaux blocs tant qu'ils n'ont pas reçu ces nouveaux blocs. La latence pousse donc naturellement au regroupement des mineurs. Elle réduit également la sécurité globale du système par le gaspillage de la recherche d'une preuve de travail valide. Le but de mettre une période de 10 minutes entre chaque bloc est une mesure pour réduire l'impact de la latence sur le système.

Termes associés : **DIFFUSION** **MINAGE**

LBTC - LIQUID BITCOIN

SIDECHAIN

Symbole boursier (*ticker*) qui désigne le bitcoin circulant sur le Liquid Network, la sidechain fédérée développée par Blockstream. Chaque L-BTC est créé par un processus d'ancrage bilatéral (*peg-in*) : des BTC sont verrouillés sur la chaîne principale dans une adresse contrôlée par la fédération, et un montant équivalent de L-BTC est émis sur Liquid. L'opération inverse (*peg-out*) détruit les L-BTC sur Liquid et libère les BTC correspondants sur la chaîne principale.

Termes associés : **BTC** **LIQUID NETWORK**

LCB/FT

ÉCONOMIE ET RÉGULATION | EN : AML/CFT

La lutte contre le blanchiment des capitaux et le financement du terrorisme (LCB/FT) fait référence aux mesures réglementaires adoptées pour prévenir l'utilisation de Bitcoin dans des activités illégales. Ces mesures incluent l'identification et la vérification de l'identité des clients (KYC), la surveillance des transactions pour détecter des schémas « suspects », et la collaboration avec les autorités pour signaler des activités considérées comme illégales. Les plateformes d'échange régulées sont tenues de s'y conformer pour opérer dans de nombreuses juridictions, notamment en France.

Termes associés : **AML - ANTI-MONEY LAUNDERING** | **ANALYSE DE CHAINE**

LDK - LIGHTNING DEV KIT

LIGHTNING NETWORK

Kit de développement (SDK) pour Lightning. LDK est une collection de bibliothèques et d'outils destinés aux développeurs pour intégrer facilement Lightning à leurs logiciels ou pour créer des applications Lightning en réduisant la complexité. LDK gère les aspects complexes de l'intégration de fonctionnalités liées à Lightning. Ce projet a été lancé par Spiral, une entreprise créée par Jack Dorsey, et s'est établi sur Rust-Lightning (RL).

Termes associés : **LIGHTNING NETWORK** | **LND**

LEDGER

PROTOCOLE | FR : GRAND LIVRE / REGISTRE

Nom parfois utilisé pour désigner le registre public et distribué qui enregistre toutes les transactions effectuées sur Bitcoin, c'est-à-dire, la blockchain. La publication et l'enregistrement des transactions sur le *ledger* permet de prévenir la double dépense sur Bitcoin, en s'assurant que chaque pièce qu'une transaction tente de dépenser n'a pas déjà été utilisée auparavant dans une autre transaction.

Termes associés : **BLOCKCHAIN** | **DOUBLE DÉPENSE**

LEDGER - ENTREPRISE

ORGANISATION

Entreprise française fondée en 2014 qui fabrique des *hardware wallets* pour sécuriser les clés privées de bitcoins. Ses appareils embarquent une puce de sécurité certifiée (*Secure Element*) qui isole les clés de tout environnement connecté à Internet.

Termes associés : **HARDWARE WALLET** | **LEDGER**

LEVELDB

PROTOCOLE

Bibliothèque de stockage de clés-valeurs légère, rapide et open-source, conçue par Google. On l'utilise sur Bitcoin pour stocker l'UTXO set et l'index des blocs. Ce système de base de données a été introduit en 2012 dans le cadre de la Pull Request « *Ultraprune* » visant à remplacer BerkeleyDB. Ce changement a eu des répercussions significatives, notamment un hard fork non intentionnel avec une réorganisation majeure de 24 blocs le 12 mars 2013. Cet incident a été détaillé dans le BIP-0050. Par la suite, la règle temporaire de limitation des *locks* introduite dans la version 0.8.1 a expiré le 15 mai 2013, entraînant un hard fork planifié.

Termes associés : **HARD FORK** **UTXO SET**

LIANA

PORTEFEUILLE

Logiciel de gestion de portefeuille Bitcoin *open source* développé par la société Wizardsardine (fondée par Kevin Loaec et Antoine Poinsot). Liana se distingue par l'utilisation de Miniscript pour permettre des configurations de sécurité avancées et très personnalisables, notamment des mécanismes de récupération basés sur des *timelocks*. Par exemple, un utilisateur peut définir une clé principale pour les dépenses courantes, puis une ou plusieurs clés de récupération qui ne deviennent actives qu'après une période d'inactivité définie. Ce modèle permet de mettre en place des plans d'héritage ou des mécanismes de sauvegarde automatiques sans tiers de confiance. Liana est compatible avec les *hardware wallets* qui prennent en charge Miniscript.

Termes associés : **MINISCRIPT** **TIMELOCK**

LIBBITCOIN

PROTOCOLE

Ensemble de bibliothèques écrites en C++ conçues pour créer des applications en lien avec Bitcoin. Libbitcoin offre des bases indépendantes pour développer des applications mobiles, des logiciels et d'autres systèmes autour de Bitcoin. Libbitcoin dispose donc d'une architecture modulaire. L'intégralité des bibliothèques du projet sont distribuées sous licence libre AGPL.

Lancé en 2011 par une équipe de développeurs menée par Amir Taaki, Libbitcoin représente alors la deuxième implémentation complète de Bitcoin, après le client originel de Satoshi. Aujourd'hui, Eric Voskuil est le mainteneur principal et le contributeur le plus actif du projet.

Termes associés : **BITCOIN CORE** **LIBSECP256K1**

LIBERTY RESERVE

HISTOIRE

Service de monnaie numérique basé au Costa Rica, fondé par Arthur Budovsky en 2006. Il permettait de créer des comptes avec une vérification d'identité minimale et de transférer des fonds libellés en « Liberty Reserve Dollars » (LR) ou en euros. Le service est devenu populaire pour les transactions illicites en raison de l'anonymat qu'il offrait. À son apogée, Liberty Reserve comptait plus d'un million d'utilisateurs dans le monde et aurait traité environ 55 millions de transactions.

En mai 2013, le gouvernement américain a saisi le service et inculpé ses opérateurs, accusés d'avoir blanchi plus de 6 milliards de dollars. Avec la fermeture antérieure d'e-gold, cet épisode a confirmé que les systèmes de monnaie numérique centralisés restent vulnérables aux interventions étatiques. Bitcoin, grâce à son architecture décentralisée qui répartit le risque entre ses participants, vise à résister à ce type de fermeture.

Termes associés : **E-GOLD** | **ECASH - DAVID CHAUM**

LIBSECP256K1

PROTOCOLE

Bibliothèque C de haute performance et de haute sécurité pour les signatures numériques et d'autres primitives cryptographiques sur la courbe elliptique `secp256k1`. Puisque cette courbe n'a jamais été largement utilisée en dehors de Bitcoin (contrairement à la courbe `secp256r1` souvent préférée), cette bibliothèque vise à être la référence la plus complète pour son utilisation. Le développement de libsecp256k1 a été principalement orienté vers les besoins de Bitcoin, et les fonctionnalités destinées à d'autres applications peuvent être moins testées ou vérifiées. Une utilisation appropriée de cette bibliothèque nécessite une attention particulière, afin de s'assurer qu'elle convienne aux objectifs spécifiques des autres applications que Bitcoin.

La bibliothèque libsecp256k1 offre une variété de fonctionnalités, notamment :

- La signature ECDSA-secp256k1 et la vérification, ainsi que la génération de clés cryptographiques ;
- Des opérations additives et multiplicatives sur les clés secrètes et publiques ;
- La sérialisation et l'analyse des clés secrètes, des clés publiques et des signatures ;
- La signature et la génération de clés publiques à temps constant et à accès mémoire constant ;
- Et une multitude d'autres primitives cryptographiques.

Terme associé : **SECP256K1**

LIBWALLY-CORE

OUTIL

Bibliothèque cryptographique et utilitaire *open source* développée par Blockstream, conçue pour faciliter l'intégration de fonctions Bitcoin dans des applications. Elle fournit des primitives bas niveau pour la gestion des clés, la construc-

tion de transactions, l'encodage d'adresses et les opérations sur les PSBT.

Libwally-core est écrite en C pour être portable et performante, avec des *bindings* disponibles pour Python, Java, JavaScript et d'autres langages. Elle s'appuie sur `libsecp256k1` pour les opérations cryptographiques sur la courbe elliptique.

Elle est notamment utilisée comme dépendance centrale de Core Lightning (CLN) pour toutes les opérations liées aux portefeuilles et aux transactions. D'autres projets comme Green Wallet de Blockstream l'utilisent également.

Termes associés : **BITCOIN CORE** **LIBSECP256K1**

LIGHTNING LABS

ORGANISATION

Entreprise fondée en 2016 par Elizabeth Stark et Olaoluwa Osuntokun, spécialisée dans le développement d'infrastructures pour le Lightning Network. Lightning Labs est le principal mainteneur de LND, l'une des implémentations les plus répandues du protocole Lightning. L'entreprise développe également un écosystème d'outils complémentaires : Lightning Loop pour les *submarine swaps*, Lightning Pool pour la liquidité, Faraday pour l'analyse de nœud, Lightning Terminal pour l'interface graphique, et le protocole Taproot Assets pour l'émission d'actifs sur Bitcoin.

Termes associés : **LND** **LOOP**

LIGHTNING NETWORK

LIGHTNING NETWORK

Protocole de couche supérieure, construit au-dessus du protocole Bitcoin, visant à permettre des transactions rapides et à faible coût. Il permet la création de canaux de paiement entre les participants, au sein desquels les transactions peuvent être effectuées presque instantanément et avec des frais minimes, sans avoir à enregistrer chaque transaction individuellement sur la blockchain. Les canaux peuvent rester ouverts quasi indéfiniment, et ne nécessitent des transactions sur la blockchain que lors de leur ouverture et de leur clôture. Le Lightning Network vise à améliorer la scalabilité de Bitcoin et à rendre possible son utilisation pour des paiements de faible valeur.

Termes associés : **HTLC** **SCALABILITÉ**

LIGHTNING NETWORK+

OUTIL

Lightning Network+ est un service en ligne qui facilite la création de *liquidity swaps* sur le Lightning Network. Plusieurs opérateurs de nœuds se coordonnent pour ouvrir simultanément des canaux de paiement entre eux de manière circulaire, ce qui garantit à chacun de recevoir autant de capacité entrante qu'il en offre. Par exemple, 3 utilisateurs A, B et C se mettent d'accord sur le site LN+ pour un *liquidity swap* de 1M sats : A ouvre un canal de 1M sats vers B, B fait de même vers C, et C ouvre à son tour un canal vers A.

Ce système permet ainsi d'améliorer la connectivité de son nœud Lightning, tout en obtenant autant de liquidité entrante que de liquidité sortante engagée.

Termes associés : **CANAL DE PAIMENT** | **LIGHTNING NETWORK**

LIGHTNING SERVICE PROVIDER

LIGHTNING NETWORK | FR : FOURNISSEUR DE SERVICE LIGHTNING

Entreprise qui opère sur Lightning dont l'activité est de fournir des services de liquidité aux utilisateurs.

Sur Lightning, les paiements sont réalisés off-chain via des canaux bidirectionnels, dont la capacité est limitée par les fonds bloqués dans le *multisig* 2/2 on-chain. Le rôle principal d'un LSP est de faciliter la gestion de cette liquidité, soit en aidant à convertir les fonds on-chain vers Lightning, soit en ouvrant de nouveaux canaux pour augmenter la capacité entrante des utilisateurs.

Ces services sont souvent utilisés par des portefeuilles *non-custodiaux*, afin que l'utilisateur puisse recevoir immédiatement des paiements Lightning sans devoir gérer lui-même l'ouverture et la gestion des canaux. En échange, le LSP facture des frais.

Termes associés : **LIGHTNING NETWORK** | **OFF-CHAIN**

LIGHTNING TERMINAL

OUTIL

Interface graphique web développée par Lightning Labs pour administrer un nœud Lightning. Lightning Terminal offre un tableau de bord visuel permettant de consulter l'état des canaux, la liquidité, les paiements routés et les performances du nœud. Il intègre les outils Loop, Pool et Faraday dans une interface unifiée.

Lightning Terminal fonctionne via le démon *litd* et peut être accessible à distance grâce au protocole *Lightning Node Connect* (LNC), sans nécessiter de VPN ou de configuration réseau complexe. Il est conçu pour simplifier la gestion quotidienne d'un nœud LND, en rendant accessibles des opérations qui requièrent habituellement la ligne de commande.

Termes associés : **LITD** | **LNC - LIGHTNING NODE CONNECT**

LIGHTSPARK

ORGANISATION

Entreprise cofondée par David Marcus, anciennement à la tête de PayPal et du projet Libra/Diem chez Meta, avec pour mission de construire une infrastructure de paiements ouverte sur Bitcoin. Lightspark a d'abord développé des services et outils pour faciliter l'intégration du Lightning Network, notamment le protocole UMA (*Universal Money Address*) permettant des transactions interopérables entre institutions. L'entreprise a ensuite créé Spark, un protocole de couche supérieure basé sur les statechains, conçu pour pallier les limites de Lightning en matière de scalabilité des portefeuilles en self-custody et de support des stablecoins. Lightspark propose également une plateforme de *Wallet-as-a-Service* et d'émission de stablecoins exploitant les capacités de Spark.

Termes associés : **LIGHTNING NETWORK** | **SPARK**

LIMITE D'ÉMISSION

PROTOCOLE

Plafond fixé pour la masse monétaire totale de BTC, établie à environ 21 millions d'unités. Cette limite est définie par la politique d'inflation dans le protocole Bitcoin. De nouveaux BTC sont créés et distribués aux mineurs via la subvention pour la validation de chaque bloc. Cette subvention est réduite de moitié tous les 210 000 blocs, un processus connu sous le nom de « *halving* ». Cette méthode garantit que la création monétaire décroîtra progressivement jusqu'à atteindre zéro.

Termes associés : **COINBASE** | **HALVING**

LIQUID NETWORK

SIDECHAIN

Sidechain de Bitcoin conçue par Blockstream pour fournir des transactions rapides et confidentielles. Contrairement à la blockchain principale de Bitcoin, Liquid utilise un mécanisme de consensus établi sur une fédération (un groupe sélectionné d'opérateurs de nœuds, généralement des entreprises liées à Bitcoin), remplaçant ainsi le mécanisme de consensus de Nakamoto. Cette approche accélère considérablement les transactions et réduit les coûts, tout en offrant des fonctionnalités plus avancées. Liquid permet aussi l'émission d'actifs numériques, y compris des jetons représentant d'autres cryptomonnaies. Les bitcoins sur Liquid, appelés L-BTC, sont liés au BTC grâce à un système d'ancrage bilatéral reposant sur une partie de la fédération. Les participants à cette fédération sont appelés des « fonctionnaires », et ils peuvent endosser à la fois le rôle de « gardien » (*watchmen*) et de « signataire de bloc » (*blocksigner*).

Termes associés : **BLOCKSTREAM** | **SIDECHAIN**

LIQUIDITÉS

LIGHTNING NETWORK | EN : LIQUIDITY - LIGHTNING

Désigne la capacité d'un canal de paiement Lightning à faire passer des transactions entre deux nœuds en fonction des bitcoins disponibles de chaque côté du canal. Il existe deux types de liquidités : entrante et sortante. La liquidité sortante représente les fonds que le nœud peut envoyer ou transférer via le canal. La liquidité entrante représente les fonds qu'il peut recevoir ou transférer dans le sens inverse. La liquidité peut aussi désigner la somme des liquidités disponibles dans tous les canaux d'un nœud.

Termes associés : **LOOP** | **NOEUD LIGHTNING**

LIQUIDITY ADS

LIGHTNING NETWORK | FR : ANNONCES DE LIQUIDITÉ

Mécanisme du protocole Lightning Network qui permet aux opérateurs de noeuds d'annoncer leur disponibilité à fournir de la liquidité entrante en échange d'une rémunération. Un noeud qui dispose de fonds peut publier une offre sur le réseau, en précisant le montant de liquidité qu'il est prêt à allouer, la durée de l'engagement et le tarif demandé. Lorsqu'un autre noeud a besoin de capacité entrante pour recevoir des paiements, il peut accepter cette offre et ouvrir un canal financé par le fournisseur de liquidité. Ce système crée un marché ouvert et décentralisé pour la liquidité sur Lightning, ce qui résout un problème récurrent pour les nouveaux noeuds qui ne peuvent pas recevoir de paiements sans capacité entrante. Les *liquidity ads* ont été formalisées dans la spécification du protocole Lightning, notamment via le message `node_announcement` qui permet aux nœuds d'annoncer leurs offres, et le protocole de *dual funding* (`open_channel2`) qui permet de négocier l'ouverture du canal. Ce système permet d'automatiser la négociation entre les parties sans intermédiaire de confiance.

Termes associés : **LIGHTNING NETWORK** | **LIQUIDITÉS**

LIQUIDITY SWAP

LIGHTNING NETWORK | FR : ÉCHANGE DE LIQUIDITÉ

Mécanisme d'ouverture coordonnée de canaux Lightning dans lequel plusieurs opérateurs de nœuds s'engagent mutuellement à ouvrir des canaux les uns vers les autres selon une disposition circulaire. Chaque participant ouvre un canal vers le suivant dans le cercle, ce qui garantit que tous obtiennent à la fois de la liquidité entrante et sortante. Ce concept est aussi connu sous le nom de « *ring of fire* ».

La plateforme LN+ est le service le plus utilisé pour organiser ces échanges. Sur cette plateforme, les participants, généralement entre trois et cinq, s'accordent sur une capacité de canal identique. Une fois tous les canaux ouverts, le cercle forme une boucle dans laquelle chaque nœud dispose d'une connectivité améliorée au réseau. Un rééquilibrage circulaire peut ensuite être effectué pour répartir la liquidité de manière équilibrée dans les canaux.

Termes associés : **CANAL DE PAIMENT** | **LIQUIDITY ADS**

LITD

⌘ OUTIL

Démon (*daemon*) qui regroupe plusieurs outils Lightning Labs en un seul processus : LND, Loop, Pool, Faraday et Taproot Assets. En exécutant *litd*, un opérateur de nœud accède à l'ensemble de ces services sans avoir à les installer et les configurer séparément. Le daemon fournit également l'interface web Lightning Terminal pour administrer le nœud, et expose le protocole *Lightning Node Connect* (LNC), permettant un accès distant sécurisé depuis un navigateur web, sans nécessiter de configuration réseau complexe comme un VPN ou une redirection de ports.

Termes associés : **LIGHTNING TERMINAL** | **LOOP**

LITECOIN

$ ÉCONOMIE ET RÉGULATION

Altcoin créé en 2011 par Charlie Lee, un ancien ingénieur de Google. Litecoin (LTC) est un fork du code source de Bitcoin avec quelques modifications de paramètres : un temps de bloc réduit à 2,5 minutes au lieu de 10, un nombre total d'unités plafonné à 84 millions (contre 21 millions pour Bitcoin), et l'utilisation de l'algorithme de hachage Scrypt au lieu de SHA-256 pour la preuve de travail. Il a parfois servi de terrain d'essai pour des fonctionnalités ensuite adoptées par Bitcoin, comme SegWit qui a été activé sur Litecoin en mai 2017, avant d'être activé sur Bitcoin en août de la même année.

Termes associés : **ALTCOIN** | **SEGWIT**

LITTLE-ENDIAN

</> INFORMATIQUE | FR : PETIT-BOUTISTE

Format de stockage de données dans les systèmes informatiques où les octets les moins significatifs (les « petits bouts ») sont placés en premier dans l'ordre des adresses. Dans une séquence comportant plusieurs octets, l'octet ayant le plus petit poids (par exemple, les chiffres les plus à droite en hexadécimale) est stocké en premier.

Termes associés : **BIG-ENDIAN** | **OCTET**

LNBITS

⌘ OUTIL

Application open source qui se place au-dessus de n'importe quelle source de financement Lightning (LND, Core Lightning, LNbits lui-même, etc.) et permet de créer des portefeuilles virtuels isolés, chacun disposant de sa propre clé API. L'intérêt principal est de ne jamais exposer le solde total du nœud sous-jacent : chaque portefeuille ne voit et ne peut dépenser que les fonds qui lui sont alloués, ce qui permet de déléguer l'accès à des applications tierces sans risque.

LNbits adopte une architecture modulaire où le cœur reste minimal et où toutes les fonctionnalités additionnelles sont fournies par des extensions optionnelles développées par la communauté : point de vente, liens de paiement, *paywalls*, NIP-05 Nostr, *Bolt Cards*, terminaux de paiement, *crowdfunding*, etc. Les déve-

loppeurs peuvent créer et partager leurs propres extensions. LNbits supporte nativement LNURL et Nostr, fournit une API REST complète, et est accessible depuis un navigateur web.

Termes associés : **LIGHTNING NETWORK** | **NOEUD LIGHTNING**

LNC - LIGHTNING NODE CONNECT

LIGHTNING NETWORK

Protocole développé par Lightning Labs permettant de se connecter à distance à un nœud Lightning via un serveur proxy, sans exposer le nœud directement sur internet. LNC établit une connexion chiffrée de bout en bout entre le client (par exemple Lightning Terminal dans un navigateur) et le nœud, en s'authentifiant à l'aide d'une *pairing phrase*. Ce protocole élimine le besoin de configurer un VPN, d'ouvrir des ports ou de gérer des certificats TLS, ce qui simplifie considérablement l'accès distant à un nœud personnel hébergé derrière un NAT ou un pare-feu.

Termes associés : **LIGHTNING TERMINAL** | **PAIRING PHRASE**

LNCLI

OUTIL

Interface en ligne de commande fournie avec LND pour administrer un nœud Lightning. LNCLI (*Lightning Network Command Line Interface*) permet d'exécuter des appels RPC vers le daemon LND pour gérer les canaux, envoyer et recevoir des paiements, consulter les balances, ajuster les frais de routage et réaliser toutes les opérations courantes d'un nœud. Chaque commande correspond à un appel gRPC sous-jacent.

Termes associés : **GRPC** | **LND**

LND

LIGHTNING NETWORK

Sigle de « *Lightning Network Daemon* ». C'est une implémentation majeure du protocole Lightning Network écrite en langage Go. Développée par Lightning Labs, LND permet la création et la gestion de canaux de paiement et de nœuds sur le réseau Lightning.

Termes associés : **ECLAIR** | **LIGHTNING NETWORK**

LNP/BP

ORGANISATION

Association à but non lucratif suisse fondée en 2019 par Maxim Orlovsky et Giacomo Zucco. L'acronyme LNP/BP désigne « *Lightning Network Protocol / Bitcoin Protocol* ». L'association est responsable du développement et de la maintenance des standards du protocole RGB, un système de smart contracts qui fonctionne en *client-side validation* sur Bitcoin et le Lightning Network. RGB permet l'émission et le transfert d'actifs numériques et de contrats intelligents directement sur Bitcoin, en conservant les données hors chaîne pour préserver la confidentialité et la scalabilité. L'association publie et maintient un ensemble de standards techniques (LNPBP) qui encadrent le fonctionnement du protocole.

Termes associés : **CLIENT-SIDE VALIDATION** | **RGB**

LNP2PBOT

OUTIL

Bot Telegram open source qui permet d'acheter et de vendre des bitcoins en pair-à-pair (P2P) via le Lightning Network. LNP2PBot fonctionne directement dans l'application de messagerie Telegram, où les utilisateurs peuvent créer des offres d'achat ou de vente, et échanger des bitcoins contre des devises locales. Les transactions sont réalisées via le Lightning Network. Un système d'*escrow* intégré au bot sécurise les transactions : il retient temporairement les sats du vendeur jusqu'à confirmation du paiement en devise par l'acheteur.

Termes associés : **LIGHTNING NETWORK** | **PEACH**

LNURL

LIGHTNING NETWORK

Protocole de communication qui spécifie un ensemble de fonctionnalités conçues pour simplifier les interactions entre les nœuds et les clients Lightning, ainsi que les applications tierces. Ce protocole repose sur HTTP et permet de créer des liens pour diverses opérations, comme une demande de paiement, une demande de retrait, ou d'autres fonctionnalités qui permettent d'améliorer l'expérience utilisateur. Chaque LNURL est une URL encodée en bech32 avec le préfixe `lnurl`, qui, une fois scannée, déclenche une série d'actions automatiques sur le portefeuille Lightning.

Par exemple, la fonctionnalité LNURL-withdraw (LUD-03) permet de retirer des fonds depuis un service en scannant un QR code, sans avoir besoin de générer manuellement une *invoice*. Ou encore, LNURL-auth (LUD-04) permet de se connecter à des services en ligne en utilisant une clé privée sur son portefeuille Lightning à la place du mot de passe.

Termes associés : **BECH32 ET BECH32M** | **BOLT**

LOCALBITCOINS

ORGANISATION

Plateforme d'échange de bitcoins en pair à pair, fondée en juin 2012 par Jeremias Kangas en Finlande. Le service mettait en relation des acheteurs et des vendeurs de bitcoins dans le monde entier, en facilitant les transactions P2P. Les utilisateurs publiaient des annonces précisant les montants, les prix et les moyens de paiement acceptés (virements bancaires, espèces, services de paiement en ligne...). Un système de dépôt fiduciaire (*escrow*) protégeait les deux parties pendant la transaction. Ce modèle permettait d'échanger des bitcoins contre de la monnaie locale dans pratiquement tous les pays, y compris ceux dépourvus de plateformes d'échange conventionnelles.

LocalBitcoins a fermé ses portes en février 2023, en raison des conditions de marché défavorables et de la réglementation croissante.

Termes associés : **BITCOIN CORE** | **MTGOX**

LOCK

PROTOCOLE

Fichier utilisé dans Bitcoin Core pour le verrouillage du répertoire de données. Il est créé lorsque bitcoind ou Bitcoin-qt démarre pour éviter que plusieurs instances du logiciel accèdent simultanément au même répertoire de données. Le but est de prévenir les conflits et les corruptions de données. Le verrouillage repose sur un mécanisme de verrou du système d'exploitation appliqué au fichier `.lock`. Le fichier lui-même reste présent sur le disque en permanence, mais le verrou est automatiquement libéré par le système d'exploitation lorsque le processus se termine, y compris en cas d'arrêt inattendu.

Termes associés : **BITCOIN CORE** | **BITCOIND**

LOGARITHME DISCRET

CRYPTOGRAPHIE | EN : DISCRETE LOGARITHM

Le logarithme discret est un problème mathématique qui est utilisé dans certains algorithmes cryptographiques à clé publique. Dans un groupe cyclique d'ordre q, avec un générateur g, si l'on a une équation de la forme $g^x = h$, alors x est appelé le logarithme discret de h par rapport à la base g, modulo q. En termes simples, il s'agit de déterminer l'exposant x lorsqu'on connaît g, h, et q. Le logarithme discret est donc la réciproque de l'exponentielle dans un groupe cyclique fini. Cependant, pour de grandes valeurs de q, résoudre le problème du logarithme discret est considéré comme algorithmiquement difficile. Cette propriété est exploitée pour assurer la sécurité de nombreux protocoles cryptographiques, tels que le protocole de Diffie-Hellman pour l'échange de clés.

Le logarithme discret est aussi utilisé dans la cryptographie à courbes elliptiques (ECC), entre autres dans l'algorithme ECDSA (*Elliptic Curve Digital Signature Algorithm*). Dans le contexte des courbes elliptiques, le problème du logarithme discret s'étend à la recherche d'un scalaire k tel que $k \cdot G = K$, où G et K sont des points sur la courbe, et $\cdot$ représente l'opération de multiplication de points. Dans le contexte de Bitcoin, les scripts peuvent utiliser soit ECDSA,

soit le protocole de Schnorr, afin de bloquer des UTXOs. Ils reposent tous deux sur l'impossibilité de calculer le logarithme discret.

Termes associés : **COURBE ELLIPTIQUE** | **ECDSA**

LOI DE GRESHAM

$ ÉCONOMIE ET RÉGULATION | EN : GRESHAM'S LAW

Principe économique selon lequel « la mauvaise monnaie chasse la bonne ». Lorsque deux monnaies circulent simultanément avec un cours légal imposé, les agents économiques tendent à thésauriser celle qui conserve sa valeur et à dépenser en priorité celle qui se déprécie. La monnaie de moindre qualité domine alors les échanges, tandis que la monnaie supérieure disparaît de la circulation.

Ce phénomène a été formulé par le financier anglais Thomas Gresham en 1558, dans une lettre à la reine Élisabeth I, pour décrire l'état de dégradation des pièces d'argent anglaises résultant du Grand Avilissement conduit sous Henri VIII et Édouard VI (1544–1551). On l'observe historiquement chaque fois qu'un État impose une parité artificielle entre deux formes de monnaie : les pièces à forte teneur en métal précieux sont fondues, exportées ou thésaurisées, tandis que les pièces altérées continuent de circuler.

Dans le contexte de Bitcoin, certains invoquent la loi de Gresham pour expliquer pourquoi les bitcoiners préfèrent dépenser des monnaies fiat, perçues comme se dépréciant, plutôt que leurs bitcoins, perçus comme s'appréciant. Selon cette lecture, le bitcoin joue le rôle de « bonne monnaie » thésaurisée, et les monnaies fiat sont dépensées en priorité, ce qui renforce le rôle de réserve de valeur du bitcoin au détriment de son usage comme moyen de paiement.

Toutefois, cette application est contestée. Au sens strict, la loi de Gresham suppose un cours légal imposé ou un taux de change fixe entre les deux monnaies, condition absente entre le bitcoin et les monnaies fiat qui s'échangent librement sur les marchés. De plus, d'autres facteurs, plus prosaïques, expliquent pourquoi le bitcoin est peu dépensé : dans de nombreuses juridictions, chaque dépense en bitcoin constitue un événement fiscal soumis à l'impôt sur les plus-values, ce qui décourage son utilisation courante ; par ailleurs, le nombre de commerçants acceptant le bitcoin reste très limité, rendant son usage en paiement simplement peu pratique.

Termes associés : **FIAT** | **LOI DE THIERS**

LOI DE THIERS

ⓢ ÉCONOMIE ET RÉGULATION | EN : THIERS' LAW

Principe économique inverse de la loi de Gresham, selon lequel « la bonne monnaie chasse la mauvaise ». Formalisée en 1989 par l'économiste Peter Bernholz en référence aux écrits de l'historien français Adolphe Thiers sur l'hyperinflation des assignats révolutionnaires, cette loi décrit un mécanisme de marché spontané. Lorsqu'une monnaie se dégrade au point de perdre la confiance des agents économiques, ceux-ci finissent par la refuser au profit d'une monnaie perçue comme plus fiable, malgré les contraintes légales.

La loi de Thiers s'observe principalement en situation d'hyperinflation. Quand une monnaie nationale s'effondre, les lois imposant son cours légal deviennent inopérantes : les commerçants exigent d'être payés en devises étrangères ou en biens tangibles. Les exemples historiques incluent le Zimbabwe, où le dollar américain s'est substitué au dollar zimbabwéen lors de sa dollarisation en 2009, ou l'Allemagne de Weimar, où le mark s'est effondré en 1923 au point que les échanges se faisaient en devises étrangères ou par troc.

Certains bitcoiners invoquent la loi de Thiers pour soutenir que Bitcoin pourrait progressivement s'imposer comme monnaie d'échange dans les économies où la monnaie nationale est fortement dépréciée, dépassant ainsi son seul rôle de réserve de valeur.

Termes associés : **HYPERINFLATION** | **LOI DE GRESHAM**

LOOP

ϟ LIGHTNING NETWORK

Service développé par Lightning Labs conçu pour faciliter l'équilibrage de liquidités dans les canaux Lightning. Loop permet aux utilisateurs de transférer des fonds entre Bitcoin et le Lightning Network, sans avoir à fermer ou ouvrir un canal. Loop aide ainsi à optimiser sa liquidité et à réduire les frais de gestion de ses canaux.

Termes associés : **LIQUIDITÉS** | **NOEUD LIGHTNING**

LOOP IN

ϟ LIGHTNING NETWORK

Opération proposée par le service Lightning Loop qui permet de convertir des bitcoins on-chain en liquidité sortante dans un canal Lightning. L'utilisateur envoie des bitcoins sur la blockchain et reçoit l'équivalent (moins les frais) dans l'un de ses canaux Lightning via un *submarine swap*. Le *Loop In* est utile pour augmenter sa capacité d'envoi sur le Lightning Network, par exemple lorsqu'un utilisateur a épuisé sa liquidité sortante après avoir effectué de nombreux paiements et souhaite pouvoir en envoyer davantage.

Termes associés : **LOOP** | **LOOP OUT**

LOOP OUT

LIGHTNING NETWORK

Opération proposée par le service Lightning Loop qui permet de convertir de la liquidité Lightning en bitcoins on-chain. L'utilisateur effectue un paiement Lightning et reçoit l'équivalent (moins les frais) dans son portefeuille Bitcoin on-chain via un *submarine swap*. Le *Loop Out* est utile pour récupérer des fonds on-chain sans fermer de canal, ou pour rééquilibrer la liquidité en déplaçant des fonds depuis un canal saturé en sortie vers le portefeuille on-chain.

Termes associés : **LOOP** **LOOP IN**

LOTTERY MINING

↗ MINAGE | FR : MINAGE PAR LOTERIE

Forme de *solo mining* dans laquelle un mineur utilise un appareil à faible hashrate pour tenter de trouver un bloc de manière autonome, sans rejoindre de pool. La probabilité de succès est extrêmement faible, mais la récompense en cas de découverte d'un bloc est perçue intégralement par le mineur.

Les appareils utilisés sont généralement des ASIC compacts ou open-source comme le Bitaxe, dont le hashrate se situe autour de 0,5 à 1,2 TH/s selon le modèle. À cette puissance, les chances de résoudre un bloc sont d'environ une sur plusieurs millions par période de dix minutes. Certains mineurs se connectent à des services de *solo mining* comme Solo CKPool ou Braiins Solo, qui leur fournissent des *templates* de blocs et relaient les blocs valides au réseau, sans mutualiser les récompenses.

La démarche relève davantage du hobby que d'une activité rentable. Ses adeptes la justifient par la contribution à la décentralisation du hashrate et par l'attrait d'une récompense intégrale en cas de succès.

Termes associés : **BITAXE** **SOLO MINING**

LUCK

↗ MINAGE | FR : CHANCE

Indicateur utilisé dans les pools de minage pour mesurer la performance d'une pool par rapport à ses attentes théoriques. Comme son nom l'indique, il évalue la chance qu'a la pool de trouver un bloc. La luck est calculée en comparant le nombre de shares théoriquement nécessaire pour trouver un bloc valide, établi sur la difficulté actuelle de Bitcoin, au nombre réel de shares produites pour trouver ce bloc. Un nombre de shares inférieur à celui attendu indique une bonne chance, tandis qu'un nombre supérieur indique une mauvaise chance.

En mettant en rapport la difficulté sur Bitcoin avec son nombre de shares produites chaque seconde et la difficulté de chaque share, la pool peut calculer le nombre de shares qui sont théoriquement nécessaires pour trouver un bloc valide. Par exemple, supposons que théoriquement, il faut 100 000 shares pour qu'une pool trouve un bloc. Si la pool en réalité en produit 200 000 avant de trouver un bloc, sa luck est de 50 % :

```
100 000 / 200 000 = 0.5 = 50 %
```

À l'inverse, si cette pool a trouvé un bloc valide après avoir seulement produit 50 000 shares, alors sa luck est de 200 % :

```
100 000 / 50 000 = 2 = 200 %
```

La luck est un indicateur qui ne peut être actualisé qu'après la découverte d'un bloc par la pool, ce qui en fait un indicateur statique mis à jour périodiquement.

Terme associé : **SHARES**

LUD

LIGHTNING NETWORK

Acronyme de « *LNURL Document* ». Les LUD sont des documents qui définissent des protocoles standardisés utilisés dans le cadre de LNURL. Chaque LUD spécifie une fonctionnalité, comme les paiements (LUD-06), les retraits (LUD-03) ou l'authentification (LUD-04). Ils permettent donc d'avoir une bonne interopérabilité entre les différents services et clients Lightning utilisant LNURL.

Terme associé : **LNURL**

LUXOR

MINAGE

Entreprise américaine fondée en 2017 par Nick Hansen, spécialisée dans les services d'infrastructure pour le minage de Bitcoin. Luxor opère une pool, un firmware personnalisé pour ASIC (LuxOS), une plateforme de données analytiques (Hashrate Index), un desk de produits dérivés sur le hashrate et une place de marché pour machines ASIC. L'entreprise cible principalement les mineurs professionnels et institutionnels nord-américains.

Terme associé : **MINING POOL**

M

M-OF-N

PORTEFEUILLE | FR : M-DE-N

Désigne un portefeuille ou un script multisignatures à seuil. Pour renforcer la sécurité de bitcoins, on peut utiliser un système de sécurisation multisignatures à seuil qui exige que `m` parmi `n` signatures soient faites pour pouvoir dépenser les fonds. Dans un m-de-n, la lettre `m` désigne le seuil de signatures requis et la lettre `n` désigne le nombre total de clés existantes pouvant signer. Par exemple, dans une configuration 2-de-3, deux signatures sur trois possibles sont nécessaires pour exécuter une transaction.

Termes associés : **MULTISIG** | **P2MS**

MACAROON

</> INFORMATIQUE

Mécanisme d'authentification conçu pour sécuriser l'accès à des services sur des systèmes distribués. Les macaroons sont notamment utilisés sur Lightning pour authentifier les utilisateurs lorsqu'ils accèdent à des services délégués. Par exemple, avec un nœud Lightning, il est possible de générer un macaroon qui autorise la réalisation de transactions à partir de votre smartphone via votre nœud distant. À la différence des *cookies*, les macaroons offrent l'avantage de pouvoir être validés cryptographiquement par l'émetteur ou d'être délégués pour vérification.

Termes associés : **LIGHTNING NETWORK** | **NOEUD LIGHTNING**

MACHANKURA

OUTIL

Machankura est un service qui permet d'envoyer et de recevoir des bitcoins par téléphone, sans connexion internet, grâce au protocole USSD. Créé en 2022 par Kgothatso Ngako, développeur sud-africain, le service rend Bitcoin accessible aux utilisateurs de téléphones basiques en Afrique. Pour effectuer une transaction, l'utilisateur compose un code USSD spécifique à son pays (par exemple `*920*8333#` au Ghana) et interagit avec un menu textuel. Les paiements transitent par le Lightning Network, et les adresses Lightning prennent la forme `numéro@8333.mobi`. Machankura est disponible dans plusieurs pays africains, dont le Nigeria, le Ghana, le Kenya et l'Afrique du Sud.

Terme associé : **LIGHTNING NETWORK**

MAGIC NETWORK

RÉSEAU

Constantes utilisées dans le protocole Bitcoin pour identifier le réseau spécifique (mainnet, testnet, regtest...) d'un message échangé entre des nœuds. Ces valeurs sont inscrites au début de chaque message pour faciliter leur identification dans le flux de données. Les Magic Network sont conçus pour être rarement présents dans des données de communication ordinaires. En effet, ces 4 octets sont peu fréquents dans l'ASCII, sont invalides en UTF-8 et génèrent un très grand entier de 32 bits, peu importe le format de stockage des données. Les Magic Network sont (en format *little-endian*) :

Mainnet :

```
f9beb4d9
```

Testnet :

```
0b110907
```

Regtest :

```
fabfb5da
```

Ces 4 octets sont parfois également nommés « Magic Number », « Magic Bytes » ou encore « Start String ».

Termes associés : REGTEST TESTNET

MAGICAL BITCOIN

PROTOCOLE

Ancien nom de la collection d'outils et de bibliothèques pour développeurs BDK.

Terme associé : BDK - BITCOIN DEV KIT

MAGMA

LIGHTNING NETWORK

Place de marché de liquidité pour le Lightning Network développée par Amboss Technologies. Magma permet aux opérateurs de nœuds d'acheter ou de vendre de la capacité entrante sous forme de canaux de paiement, indépendamment de l'implémentation utilisée (LND, CLN, Eclair).

Le mécanisme repose sur des *HODL invoices* pour créer des contrats conditionnels : le vendeur ne reçoit le paiement que s'il ouvre effectivement le canal convenu dans un délai court. Ce système permet à Magma de faciliter les échanges sans jamais prendre la garde des fonds des utilisateurs.

Magma répond à un besoin récurrent sur le Lightning Network : obtenir de la liquidité entrante. Un nœud qui vient d'ouvrir un canal ne dispose que de capacité sortante ; pour recevoir des paiements, il doit convaincre un pair d'allouer des fonds de son côté du canal. Magma transforme ce processus en un marché ouvert où la liquidité s'échange contre des frais négociés entre les parties.

Termes associés : AMBOSS HODL INVOICE

MAINNET

PROTOCOLE

Désigne le réseau principal où les transactions réelles de Bitcoin sont enregistrées et exécutées. Le *mainnet* est tout simplement le réseau Bitcoin. Contrairement aux *testnets*, *regtests* et *signets*, le *mainnet* implique l'utilisation de bitcoins ayant une valeur économique réelle.

Termes associés : **SIGNET** **TESTNET**

MAINTENEUR - CORE

COMMUNAUTÉ | EN : MAINTAINER

Dans le contexte du projet Bitcoin Core, l'implémentation majoritaire de nœuds sur le réseau Bitcoin, les mainteneurs sont des individus chargés de la gestion du projet. Ils portent la responsabilité de l'implémentation. Ils sont chargés de la modération sur le dépôt GitHub Bitcoin Core et de l'établissement du calendrier pour la publication des nouvelles versions. Ils sont surtout chargés de conduire la fusion des *pull requests* (PR) proposées par les contributeurs.

Autrement dit, lorsqu'une proposition de modification du code a passé les divers stades de validation, ce sont ces mainteneurs qui assument la grande responsabilité de fusionner le nouveau code avec le logiciel Bitcoin Core. Avant de procéder à cette fusion, les mainteneurs vérifient si le code respecte bien les principes fondamentaux du projet, s'il a atteint les standards minimums requis pour être inclus, et ils jugent également le consensus général des contributeurs à propos de cette modification. Lorsque j'écris cette définition (mars 2026), il y a cinq mainteneurs sur Bitcoin Core : Hennadii Stepanov, Michael Ford, Ava Chow, Ryan Ofsky et Sedited.

Termes associés : **BITCOIN CORE** **CONTRIBUTEUR - CORE**

MAINTENEUR PRINCIPAL - CORE

COMMUNAUTÉ | EN : LEAD MAINTAINER

Le mainteneur principal était un rôle au sein de la hiérarchie sur Bitcoin Core. Ce rôle n'existe plus depuis février 2023. Cette personne était chargée de diriger le projet et avait donc plus de pouvoir que les mainteneurs. Le rôle de mainteneur principal fut naturellement endossé par Satoshi Nakamoto jusqu'à son départ au début de l'année 2011. Par la suite, Gavin Andresen, ayant déjà contribué aux côtés de Satoshi, prit la relève à la tête du logiciel jusqu'au début de l'année 2014. À partir de cette date, Wladimir J. van der Laan a pris ce rôle jusqu'en février 2023. Depuis, il n'y a plus aucun mainteneur principal pour le projet.

Termes associés : **BITCOIN CORE** **MAINTENEUR - CORE**

MAJORITÉ ÉCONOMIQUE

PROTOCOLE | EN : ECONOMIC MAJORITY

Désigne la plus grande proportion de l'activité économique liée à la monnaie bitcoin, générée par les commerçants. Un commerçant désigne toute entité physique ou morale acceptant d'échanger un bien ou un service contre du BTC. Ces commerçants, qui incluent les commerces, les utilisateurs, les plateformes d'échange, et les mineurs, varient en taille et en influence économique. Certains sont des acteurs majeurs, générant une activité économique substantielle, tandis que d'autres sont plus modestes. La majorité économique est donc définie par ceux dont l'activité économique combinée représente la part prépondérante sur cette monnaie. Cette majorité a une influence sur les règles de consensus, notamment en cas de fork.

Termes associés : **CONSENSUS** | **FORK**

MALLÉABILITÉ - TRANSACTION

PROTOCOLE | EN : TRANSACTION MALLEABILITY

Se réfère à la possibilité de modifier légèrement la structure d'une transaction Bitcoin, sans en altérer l'effet, mais tout en changeant l'identifiant de transaction (*TXID*). Cette propriété peut être exploitée de manière malveillante pour induire en erreur les parties prenantes sur le statut d'une transaction, causant ainsi des problèmes comme la double dépense. La malléabilité était rendue possible par la flexibilité de la signature numérique utilisée. La malléabilité rendait notamment compliquée une implémentation du Lightning Network. Le soft fork SegWit a été introduit pour résoudre ce problème, en écartant les données malléables de la transaction du calcul du TXID.

Bien que ce soit rare, on retrouve parfois le terme de « mutabilité » pour évoquer le même concept.

Termes associés : **SEGWIT** | **TXID - TRANSACTION IDENTIFIER**

MAPPER

</> INFORMATIQUE | EN : TO MAP

Dans le contexte de l'informatique, mapper désigne le processus d'associer des éléments d'un ensemble de données à des éléments d'un autre ensemble de données de manière systématique. Cette association permet aux données du premier ensemble de se substituer à celles du second ensemble ou de transitionner de l'un à l'autre. Cette technique est souvent utilisée dans les opérations de transformation de données.

Termes associés : **ALGORITHME** | **FONCTION DE HACHAGE**

MARA

↗ MINAGE

Entreprise américaine de minage de Bitcoin cotée au NASDAQ sous le symbole MARA. Fondée en 2010 et initialement active dans d'autres secteurs, elle a pivoté vers le minage de Bitcoin à partir de 2017, alors sous le nom de Marathon Patent Group. En 2021, elle a été renommée Marathon Digital Holdings, puis MARA Holdings en 2024. L'entreprise exploite des centres de données dans plusieurs États américains et figure parmi les plus grandes sociétés de minage au monde en termes de hashrate. MARA opère également sa propre pool de minage, MARAPool.

Terme associé : **MINING POOL**

MARKET CAP

$ ÉCONOMIE ET RÉGULATION | FR : CAPITALISATION BOURSIÈRE

Représente la valeur totale d'un actif en circulation, comme le bitcoin. Elle est calculée en multipliant le nombre total de pièces en circulation par le prix actuel de chaque unité. Ce chiffre donne une indication de la taille globale et de la valeur du marché de Bitcoin.

Termes associés : **BEAR MARKET** **BULL MARKET**

MASF

⚙ PROTOCOLE

Sigle de « *Miner-Activated Soft Fork* ». Qualifie un soft fork dans Bitcoin lorsque son activation provient d'une action des mineurs. Les MASF sont une famille de méthodes d'activation de soft fork sur Bitcoin. Dans ces approches, les mineurs signalent leur accord et leur préparation pour une mise à jour du protocole en minant des blocs qui soutiennent le verrouillage du soft fork. Si une majorité significative de mineurs se prononce en faveur du soft fork, la mise à jour est considérée comme acceptée et est activée ultérieurement. Ce processus permet d'éviter la division de la blockchain et de maintenir l'unité du réseau. Le MASF est préféré pour son approche plus douce et consensuelle, réduisant le risque de scission de la blockchain tout en assurant que la majorité de la puissance de calcul soutient la nouvelle mise à jour. Les méthodes d'activation BIP-0034, BIP-0009, BIP-0008 (si `LOT=false` ou si le seuil de vote est atteint) ou encore Speedy Trial sont des MASF.

Termes associés : **SOFT FORK** **UASF**

MAST

CRYPTOGRAPHIE

Sigle de « *Merklized Alternative Script Trees* ». Technique employant un arbre de Merkle pour résumer un nombre arbitraire de conditions de dépenses sélectionnées par l'utilisateur dans une adresse de réception, dont une doit être remplie pour dépenser les bitcoins concernés. L'arbre de Merkle permet à l'utilisateur de choisir quelle condition il souhaite remplir sans révéler les détails des autres conditions sur la blockchain. Cela permet de réduire les frais liés à ces scripts, de créer des conditions beaucoup plus lourdes et, sur un temps plus long, d'améliorer la confidentialité de l'utilisateur (en plus de l'utilisation conjointe de Schnorr). Ce concept a fait l'objet de plusieurs propositions, mais il a finalement été ajouté à Bitcoin via le soft fork Taproot en 2021.

Initialement, « MAST » était l'acronyme de « Merklized Abstract Syntax Tree ». L'utilisation qui en est faite dans le cadre de Taproot n'a plus rien à voir avec un « Abstract Syntax Tree ». Toutefois, les utilisateurs continuaient d'employer ce terme de MAST. Anthony Towns a donc proposé de modifier la signification initiale tout en conservant cet acronyme largement employé avec : « Merklized Alternative Script Trees ».

Termes associés : **ARBRE DE MERKLE** | **TAPROOT**

MASTER FINGERPRINT

PORTEFEUILLE | FR : EMPREINTE MAÎTRESSE

Empreinte de 4 octets (32 bits) de la clé privée maîtresse dans un portefeuille hiérarchique déterministe (HD). Elle est obtenue en calculant le hash `SHA256` de la clé privée maîtresse, suivi d'un hash `RIPEMD160`, procédé désigné par `HASH160` sur Bitcoin. La Master Fingerprint sert à identifier un portefeuille HD, indépendamment des chemins de dérivation, mais en prenant en compte la présence ou non d'une passphrase. C'est une information concise qui permet de faire référence à l'origine d'un ensemble de clés, sans pour autant dévoiler des informations sensibles sur le portefeuille.

Termes associés : **CHEMIN DE DÉRIVATION** | **HASH160**

MATT

SCRIPT

Proposition de soft fork dont le nom signifie « *Merkleize All The Things* » et qui vise à introduire des capacités de covenant et d'introspection limitée des transactions dans Bitcoin. MATT a été proposé par Salvatore Ingala en 2022.

Le cœur de la proposition repose sur l'opcode `OP_CHECKCONTRACTVERIFY` (abrégé `OP_CCV`), qui permet de vérifier qu'une sortie de transaction s'engage sur un programme et sur des données spécifiques au moyen d'un arbre de Merkle. Concrètement, chaque état d'un contrat est représenté par un nœud dans un arbre de Merkle, et les transitions entre états sont validées par le script lui-même. Cette structure rend possible la vérification de programmes arbitraires dans des protocoles contractuels, sans nécessiter de modification du consensus pour chaque nouveau type de contrat.

MATT pourrait permettre l'implémentation de fonctionnalités telles que les coffres-forts (*vaults*), les canaux de paiement améliorés ou d'autres constructions fondées sur les covenants. La proposition `OP_VAULT`, qui visait initialement un mécanisme de coffre-fort dédié, a d'ailleurs été retirée en faveur de `OP_CCV`, jugé plus généraliste. L'opcode a été formellement intégré dans la documentation des BIP en 2025.

Termes associés : **SCHNORR** | **SOFT FORK**

MAX_BLOC_SIZE

PROTOCOLE

Constante qui spécifie la taille maximale qu'un bloc peut avoir sur Bitcoin. Historiquement, cette limite était fixée à 1 Mo, une mesure mise en place par Satoshi Nakamoto en 2010, afin de prévenir le spam et de maintenir une certaine décentralisation du réseau.

Termes associés : **BLOC** | **BLOCKSIZE WAR**

MAXIMALISTE

COMMUNAUTÉ

Terme qui désigne une personne convaincue que Bitcoin est la seule cryptomonnaie dotée d'une valeur fondamentale, et que les altcoins sont superflus, voire nuisibles. Le mot est issu de l'anglais *Bitcoin maximalist*, expression forgée par Vitalik Buterin dans un article de blog publié en novembre 2014. Initialement employée de manière péjorative pour critiquer une vision jugée étroite de l'écosystème des cryptomonnaies, l'étiquette a ensuite été revendiquée par ceux qu'elle désignait.

Termes associés : **ALTCOIN** | **BITCOIN - B MAJUSCULE** | **SHITCOIN**

MÉLANGEUR

CONFIDENTIALITÉ | EN : MIXER

Service centralisé permettant de casser l'historique de pièces Bitcoin en mélangeant les fonds de plusieurs utilisateurs. Contrairement au coinjoin, où les utilisateurs conservent le contrôle de leurs fonds tout au long du processus de mixage, un mélangeur prend temporairement possession des bitcoins des utilisateurs, les mélange avec ceux d'autres utilisateurs, puis renvoie les fonds sur de nouvelles adresses.

De nos jours, l'usage des mélangeurs de bitcoins s'estompe au profit des coinjoins chaumiens, qui offrent plus de sécurité en éliminant le besoin de confiance. Contrairement aux mélangeurs, où les utilisateurs doivent confier leurs bitcoins à un opérateur pouvant les subtiliser ou conserver des données sur le processus de mixage, les coinjoins ne présentent pas ces risques.

Terme associé : **COINJOIN**

MEMPOOL

PROTOCOLE

Contraction des termes « memory » et « pool ». Cela désigne un espace virtuel dans lequel les transactions Bitcoin en attente d'inclusion dans un bloc sont regroupées. Lorsqu'une transaction est créée et diffusée sur le réseau Bitcoin, elle est d'abord vérifiée par les nœuds du réseau. Si elle est considérée comme valide, elle est alors placée dans la Mempool de chaque nœud, où elle reste jusqu'à ce qu'elle soit sélectionnée par un mineur pour être incluse dans un bloc.

Il est important de noter que chaque nœud du réseau Bitcoin maintient sa propre Mempool, et donc, il peut y avoir des variations dans le contenu de la Mempool entre différents nœuds à un moment donné. Notamment, le paramètre `maxmempool` dans le fichier `bitcoin.conf` de chaque nœud permet aux opérateurs de contrôler la quantité de RAM (mémoire vive) que leur nœud utilisera pour stocker les transactions en attente dans la Mempool. En limitant la taille de la Mempool, les opérateurs de nœuds peuvent éviter que celle-ci ne consomme trop de ressources sur leur système. Ce paramètre est spécifié en mégaoctets (MB). La valeur par défaut de Bitcoin Core à ce jour est de 300 Mo.

Les transactions présentes dans les mempools sont provisoires. Elles ne doivent pas être considérées comme immuables tant qu'elles ne sont pas incluses dans un bloc, et après un certain nombre de confirmations. Celles-ci peuvent souvent être remplacées ou purgées.

Termes associés : **BITCOIN.CONF** **TRANSACTION NON CONFIRMÉE**

MEMPOOL.DAT

PROTOCOLE

Nom du fichier de données utilisé par le logiciel Bitcoin Core pour stocker l'état actuel de la mempool, qui est l'ensemble des transactions non confirmées en attente d'être ajoutées à un bloc.

Termes associés : **BITCOIN CORE** **MEMPOOL**

MEMPOOL.SPACE

OUTIL

Explorateur de blocs et visualiseur de mempool open source pour Bitcoin, accessible à l'adresse mempool.space. Cet outil permet de consulter en temps réel l'état de la mempool (les transactions en attente de confirmation), d'estimer les frais de transaction optimaux, d'explorer les blocs, les transactions et les adresses, et de visualiser l'état du minage. Mempool.space peut également être auto-hébergé sur son propre nœud pour une utilisation souveraine et privée.

Termes associés : **BLOCKCHAIN** **UTXO**

MERGE

‹/› INFORMATIQUE | FR : FUSIONNER

Dans le cadre de Git, représente l'action d'intégrer les modifications d'une branche à une autre, typiquement de ramener les développements d'une branche secondaire dans la branche principale. Cette opération permet de combiner les historiques de commit des branches concernées et de résoudre les éventuels conflits pour maintenir l'intégrité du logiciel.

Termes associés : **BRANCHE - GIT** | **GIT**

MERKLE BLOCK

✿ PROTOCOLE

Structure de données utilisée dans le cadre du BIP-0037 (*Transaction Bloom Filtering*) pour fournir une preuve compacte de l'inclusion de transactions spécifiques dans un bloc. C'est notamment utilisé pour les portefeuilles SPV. Le Merkle Block contient l'en-tête de bloc, les transactions filtrées et un arbre de Merkle partiel, permettant aux clients légers de vérifier rapidement si une transaction appartient à un bloc sans télécharger toutes les transactions.

Termes associés : **ARBRE DE MERKLE** | **BIP-0037**

MERKLE SUM TREE

‹/› INFORMATIQUE

Variante d'un arbre de Merkle dans laquelle chaque nœud porte non seulement le hash des nœuds ou feuilles situés en dessous, mais également la somme de leurs valeurs. La racine de l'arbre contient ainsi le hash de l'ensemble des données et la somme totale de toutes les valeurs des feuilles. Cette structure permet de vérifier simultanément l'intégrité des données et la cohérence des montants. Dans le protocole Taproot Assets, les *Merkle Sum Trees* sont utilisés pour engager les soldes d'actifs dans la blockchain Bitcoin, garantissant qu'aucun actif ne peut être créé ou détruit sans modifier la racine de l'arbre.

Termes associés : **ARBRE DE MERKLE** | **TAPROOT ASSETS PROTOCOL**

MESURE À LA PRISE

↗ MINAGE | EN : MEASURE AT THE WALL

Dans le cadre de l'industrie du minage, désigne la mesure de la consommation électrique réelle d'un ASIC, directement à son point de connexion, c'est-à-dire à la prise électrique murale. Contrairement aux spécifications théoriques, cette mesure permet de quantifier précisément l'énergie consommée par la machine en incluant toutes les pertes lors du fonctionnement réel.

Termes associés : **ASIC** | **HASHRATE**

MÉTADONNÉES

</> INFORMATIQUE | EN : METADATA

Dans le domaine général de l'informatique, cela désigne les données qui fournissent des informations sur d'autres données. Elles décrivent les caractéristiques, le contenu, la qualité, le format et la structure des données qu'elles accompagnent. On différencie ainsi la charge utile (payload), qui représente le cœur de l'information, et les métadonnées. Par exemple, pour un document, les métadonnées peuvent inclure l'auteur, la date de création, la taille du fichier et les mots-clés associés. Sur Bitcoin, on retrouve des métadonnées dans de nombreux éléments. On en utilise dans les adresses de réception, les clés étendues, les blocs...

Termes associés : **ADRESSE DE RÉCEPTION** | **CLÉ ÉTENDUE**

MÉTHODE D'ACTIVATION

⚙ PROTOCOLE | EN : ACTIVATION METHOD

Une méthode d'activation est le processus par lequel la communauté d'utilisateurs décide de l'implémentation d'un soft fork sur le protocole Bitcoin, en cherchant à éviter une séparation de la blockchain. Ce processus consiste à solliciter l'opinion des mineurs pour approuver un soft fork avant son activation. Si une majorité importante accepte le soft fork, le risque de scission de la blockchain est minimisé. Ce consensus est crucial car si une majorité de mineurs refusent de faire la modification, le soft fork pourrait créer deux chaînes distinctes : une avec les règles modifiées et l'autre sans. Il existe 2 grandes catégories de méthodes d'activation :

- Les UASF (User-Activated Soft Fork) lorsque ce sont les nœuds qui imposent la mise à jour ;
- Les MASF (Miner-Activated Soft Fork) lorsque ce sont les mineurs qui déclenchent l'activation.

Il existe de nombreuses méthodes d'activation différentes qui ont été testées au fur et à mesure de l'évolution de Bitcoin. À l'époque de Satoshi, le processus d'activation n'était pas formellement établi. Les modifications étaient souvent arbitraires et parfois même réalisées sans informer la communauté. Plus tard, la méthode du *Flag Day* a été adoptée. Après le retrait de Satoshi, d'autres méthodes ont été successivement utilisées, notamment le BIP-0034, le BIP-0009, le BIP-0008, et enfin, le *Speedy Trial*.

Termes associés : **SOFT FORK** | **UASF**

MÉTHODE GÉOMÉTRIQUE

↗ MINAGE | EN : GEOMETRIC METHOD

Méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. Ce système de paiement est établi sur un score, conçu pour contrer le phénomène de *pool hopping*. Elle assure que le paiement par share soumise reste constant, indépendamment du moment de soumission. Les mineurs accumulent des scores, calculés avec un facteur de décroissance, et les paiements sont calculés à la fin du cycle. Ils sont proportionnels à leur score. Cette mé-

thode implique des frais variables et fixes pour le mineur et réduit la variance des paiements par share.

Terme associé : **SHARES**

MILLIBITCOIN

PROTOCOLE

Sous-unité du bitcoin équivalant à un millième de bitcoin, soit 0,001 BTC ou 100 000 satoshis. Le millibitcoin est parfois abrégé en « mBTC ». Cette unité est utilisée pour exprimer des montants intermédiaires de manière plus lisible que les fractions décimales de bitcoin, notamment dans les interfaces de portefeuilles et les plateformes d'échange.

Terme associé : **MILLISATOSHI**

MILLISATOSHI

LIGHTNING NETWORK

Sous-unité du satoshi équivalant à un millième de satoshi, soit un cent-milliardième de bitcoin (10^{-11} BTC). Le millisatoshi n'existe pas sur la blockchain Bitcoin, où le satoshi est l'unité indivisible minimale. En revanche, le Lightning Network utilise le millisatoshi comme unité de compte interne pour les paiements et le calcul des frais de routage, permettant une granularité plus fine.

Termes associés : **LIGHTNING NETWORK** **MILLIBITCOIN**

MIMBLEWIMBLE

CONFIDENTIALITÉ

Protocole qui vise à améliorer la confidentialité et le passage à l'échelle des transactions. Le *white paper* de ce protocole a été déposé en 2016 sur le canal IRC `#bitcoin-wizards` par un utilisateur sous pseudonyme. Andrew Poelstra a ensuite formalisé le protocole dans un article publié en octobre 2016. Mimblewimble repose sur les *Confidential Transactions* et les engagements de Pedersen pour masquer les montants, sans empêcher la vérification que les entrées égalent les sorties. Il n'utilise ni adresses, ni système de scripts : l'authentification se fait uniquement par la structure algébrique des transactions. Toutes les transactions d'un bloc sont agrégées en une seule, à la manière d'un *coinjoin* géant, ce qui rend les liens entre émetteur et destinataire indistinguables.

Le protocole a été mis en œuvre dans deux blockchains distinctes lancées en janvier 2019 : Grin et Beam. Litecoin a intégré une version adaptée sous forme de blocs d'extension en mai 2022. Mimblewimble n'a jamais été intégré à Bitcoin en raison de son incompatibilité avec le système de scripts et le modèle d'UTXO actuels.

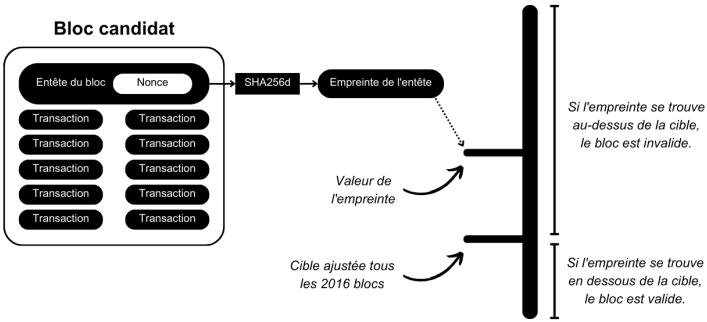
Termes associés : **CONFIDENTIAL TRANSACTIONS** **LITECOIN** **PEDERSEN COMMITMENT**

MINAGE

↗ MINAGE | EN : MINING

Action de participer à la preuve de travail (*Proof-of-Work*) du système Bitcoin. La preuve de travail est un mécanisme de résistance aux attaques Sybil. Elle est à la base du mécanisme de consensus de Nakamoto, qui est le principe utilisé pour établir un accord sur une version unique du registre distribué entre les différents nœuds du réseau.

Concrètement, le minage est la recherche d'une valeur qui, une fois passée dans une fonction mathématique aléatoire, donne un résultat inférieur à un nombre cible. Cette cible de la preuve de travail est ajustée tous les 2016 blocs par les nœuds. C'est ce que l'on appelle l'ajustement de la difficulté. On abaisse le nombre cible pour augmenter la difficulté de minage, ou on l'augmente pour baisser la difficulté, en fonction de l'évolution de la puissance de calcul déployée par les mineurs durant la période précédente.



Ce travail effectué par les mineurs est récompensé à chaque bloc valide trouvé. Le mineur gagnant empoche une récompense pécuniaire, composée de la subvention de bloc (création de nouveaux bitcoins ex-nihilo), et des frais de transaction. Aujourd'hui, la difficulté de la preuve de travail sur Bitcoin est telle que le minage nécessite une grande puissance de calcul pour parvenir à gagner des blocs. En conséquence, il faut souvent disposer de puces électroniques spécialisées dans l'exécution de `SHA256d`, que l'on appelle des ASICs, et participer à des pools de minage.

Termes associés : **ASIC** | **PREUVE DE TRAVAIL**

MINAGE FUSIONNÉ

SIDECHAIN | EN : MM - MERGED MINING

Technique de consensus de sidechain permettant aux mineurs de Bitcoin de travailler simultanément sur la chaîne principale et sur une ou plusieurs sidechains, sans pour autant devoir fournir plus de travail de calcul. Il s'agit donc de réutiliser la preuve de travail de Bitcoin pour des applications tierces. Toutefois, le minage fusionné présente un désavantage notable pour le mineur : il nécessite l'installation et l'exécution d'un logiciel de nœud spécifique à chaque sidechain pour permettre la réutilisation de ses preuves de travail. De plus, la récompense obtenue pour le minage d'une sidechain est versée sur celle-ci et non directement en BTC sur la blockchain principale.

Termes associés : **MINAGE FUSIONNÉ AVEUGLE** | **SIDECHAIN**

MINAGE FUSIONNÉ AVEUGLE

SIDECHAIN | EN : BMM - BLIND MERGED MINING

Technique de consensus de sidechain permettant aux mineurs de Bitcoin de travailler simultanément sur la chaîne principale et sur une ou plusieurs sidechains, sans pour autant devoir fournir plus de travail de calcul. Contrairement au minage fusionné classique, cette méthode ne nécessite pas de configurer un nouveau nœud pour chaque sidechain exploitée. Dans le cadre du *Blind Merged Mining* (BMM), chaque sidechain est gérée par des opérateurs de nœud indépendants, responsables de la création des blocs et de la récolte des récompenses associées sur la sidechain. En contrepartie, ces opérateurs doivent acheter des preuves de travail auprès des mineurs de la blockchain principale pour valider leurs blocs sur la sidechain. Ainsi, les mineurs de Bitcoin reçoivent leurs récompenses du minage fusionné des sidechains en BTC, directement sur la chaîne principale. Cette méthode, développée par Paul Sztorc pour les drivechains, nécessite l'implémentation du BIP-0301 pour être opérationnelle sur le réseau Bitcoin.

Termes associés : **DRIVECHAIN** | **MINAGE FUSIONNÉ**

MINEUR

↗ MINAGE | EN : MINER

Dans le contexte de Bitcoin, un mineur fait référence à une personne qui gère une mine, c'est-à-dire un ordinateur engagé dans le processus de minage, qui consiste à participer à la preuve de travail (*Proof-of-Work*). Le mineur regroupe les transactions en attente dans sa mempool pour former un bloc candidat. Ensuite, il recherche un hachage valide, inférieur ou égal à la cible, pour l'entête de ce bloc en modifiant les différents nonces. S'il trouve un hachage valide, il diffuse son bloc au réseau Bitcoin et empoche la récompense pécuniaire associée, composée de la subvention de bloc (création de nouveaux bitcoins ex-nihilo), et des frais de transaction.

Dans le cadre spécifique des pools de minage, on différencie parfois le rôle de mineur du rôle de hacheur, étant donné que les hacheurs individuels qui participent à la mutualisation doivent uniquement hacher et ne participent pas

au processus complet de la mine.

Termes associés : **MINAGE** | **MINING POOL**

MINIBITS

⊸ PORTEFEUILLE

Portefeuille mobile dédié à l'eCash, qui utilise le protocole Cashu pour émettre des jetons numériques adossés à du bitcoin via le Lightning Network. L'application permet d'envoyer et de recevoir des paiements de manière quasi instantanée, avec un haut niveau de confidentialité, puisque les transactions eCash sont difficilement traçables.

Termes associés : **CASHU** | **LIGHTNING NETWORK**

MINING MANAGEMENT SOFTWARE

↗ MINAGE | FR : LOGICIEL DE GESTION DE MINAGE

Logiciel dédié à la gestion et à l'optimisation des opérations de minage. Ce type de logiciel permet de surveiller en temps réel les performances des ASICs, la consommation énergétique, les températures, et les taux de hachage. Il offre également des fonctionnalités de gestion des fermes de minage à grande échelle, avec l'automatisation des réglages (comme l'*overclocking* par exemple), la détection des pannes et la gestion à distance des appareils.

Termes associés : **ASIC** | **FERME DE MINAGE**

MINING POOL

↗ MINAGE | FR : COOPÉRATIVE DE MINAGE

Fait référence à un ensemble de mineurs qui collaborent en combinant leur puissance de calcul pour participer à la recherche de preuves de travail valides sur Bitcoin. Cette mutualisation en une seule organisation est une solution à la difficulté croissante de l'extraction de bitcoins, qui rend trop improbable pour un mineur individuel de rivaliser et de gagner des récompenses de manière stable. Les mineurs au sein d'une pool de minage contribuent avec leur machine à la recherche de shares valides. Lorsqu'un bloc est miné par la pool, la récompense, qui comprend les bitcoins nouvellement créés ainsi que les frais de transaction inclus dans le bloc, est distribuée parmi les membres de la pool en fonction de la méthode de rémunération choisie. Cette distribution est proportionnelle à la puissance de calcul que chaque mineur a contribué.

En joignant leurs forces, les mineurs au sein d'une pool augmentent leurs chances de trouver des blocs et réduisent ainsi leur variance. Cela permet d'assurer une source de revenus plus régulière et prévisible par rapport au minage en solo, où un mineur peut ne pas gagner de récompense pendant de longues périodes. Ils peuvent ainsi lisser leurs gains, et donc avoir une meilleure visibilité sur leur activité, notamment pour faire face aux différentes charges qu'elle induit.

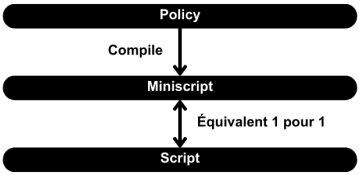
Attention, la pool de minage ne doit pas être confondue avec la ferme de minage.

Terme associé : **SHARES**

MINISCRIPT

SCRIPT

Framework permettant de fournir un cadre pour programmer des scripts de manière sécurisée sur Bitcoin. Le langage natif de Bitcoin s'appelle script. Celui-ci est assez complexe à utiliser en pratique, notamment pour des applications sophistiquées et personnalisées. Surtout, il est très difficile de vérifier les limitations d'un script. Miniscript utilise un sous-ensemble de scripts Bitcoin pour simplifier leur création, leur analyse et leur vérification. Chaque miniscript est équivalent 1 pour 1 avec un script natif. On utilise un langage de *policies* facile à utiliser, qui est ensuite compilé en miniscript, pour enfin correspondre à un script natif.



Miniscript permet ainsi aux développeurs de construire des scripts sophistiqués d'une manière plus sûre et plus fiable. Les propriétés essentielles de Miniscript sont les suivantes :

- Il permet une analyse statique du script, notamment des conditions de dépenses qu'il permet et de son coût en termes de ressources ;
- Il permet de réaliser des scripts qui respectent le consensus ;
- Il permet d'analyser si oui ou non, les différents chemins de dépense respectent les règles standards des nœuds ;
- Il permet de mettre en place un standard général, compréhensible et composable, pour l'ensemble des logiciels et matériels de portefeuille.

Le projet Miniscript a été lancé en 2018 par Pieter Wuille, Andrew Poelstra et Sanket Kanjalkar, via l'entreprise Blockstream. Miniscript est ajouté au wallet Bitcoin Core en mode *watch-only* en décembre 2022 avec la version 24.0, puis complètement en mai 2023 avec la version 25.0.

Termes associés : MINITAPSCRIPT | SCRIPT

MINISKETCH

RÉSEAU

Bibliothèque de réconciliation d'ensembles (*set reconciliation*) développée principalement par Pieter Wuille (sipa). Minisketch implémente un algorithme qui permet à deux pairs d'un réseau pair-à-pair de déterminer efficacement les éléments que l'un possède et que l'autre n'a pas, sans que chacun doive transmettre la liste complète de ses éléments.

Le principe repose sur des esquisses (*sketches*) : chaque pair calcule une représentation compacte de son ensemble d'éléments, dont la taille dépend du nombre de différences attendues et non de la taille totale des ensembles. En

échangeant ces esquisses, les deux pairs peuvent identifier les éléments manquants de chaque côté et ne transmettre que ceux-ci. Ce mécanisme réduit considérablement la bande passante consommée par les protocoles de diffusion (*gossip*).

Minisketch constitue la brique technique fondamentale d'Erlay, une proposition d'amélioration du relais de transactions sur le réseau Bitcoin. Actuellement, chaque noeud annonce chaque transaction à tous ses pairs, ce qui génère une redondance importante. Avec Erlay, les noeuds n'annonceraient les transactions qu'à un petit nombre de pairs, puis utiliseraient la réconciliation via Minisketch pour synchroniser leurs mempools avec les autres. Cette approche pourrait réduire d'environ 84 % la bande passante utilisée pour l'annonce des transactions, et d'environ 40 % la consommation globale de bande passante.

MINITAPSCRIPT

SCRIPT

Version de Miniscript pour Tapscript. Tapscript dispose de quelques différences notables avec Script dans sa version originale. MiniTapscript fournit ainsi la prise en charge de Tapscript dans Miniscript.

Ce terme est parfois contesté. En effet, certains bitcoiners préfèrent parler de « TapMiniscript ».

Termes associés : **MINISCRIPT** **TAPSCRIPT**

MIT DCI

ORGANISATION

Programme de recherche du MIT Media Lab consacré aux monnaies numériques et aux technologies de registre distribué. Fondée en avril 2015, cette initiative a été le premier groupe universitaire exclusivement dédié à Bitcoin. Sa création intervient dans un contexte où la Bitcoin Foundation, jusqu'alors principale source de financement du développement, faisait face à des difficultés financières. Le DCI finance des développeurs et des chercheurs qui travaillent sur Bitcoin Core et sur la sécurité du protocole. Il rejoint ainsi d'autres organisations comme Chaincode Labs, Blockstream ou Spiral dans le soutien au développement open source de Bitcoin.

Termes associés : **BITCOIN CORE** **CHAINCODE LABS**

MIT X11

</> INFORMATIQUE

Licence de logiciel libre très permissive qui autorise les utilisateurs à copier, modifier, fusionner, publier, distribuer, sous-licencier et vendre le logiciel. Elle exige uniquement que la licence originale et les notifications de droits d'auteur soient conservées dans toutes les copies ou distributions substantielles du logiciel. Contrairement à la licence GPL, la licence MIT ne requiert pas que les adaptations ou les versions dérivées du logiciel soient distribuées sous la même licence. Cette flexibilité fait de la licence MIT X11 un choix populaire pour de nombreux projets *open source*, y compris dans l'environnement de Bitcoin. Satoshi Nakamoto a d'ailleurs choisi cette licence pour la première version de Bitcoin publiée en 2009, et elle reste en usage pour le projet Bitcoin Core aujourd'hui.

Termes associés : **BITCOIN CORE** | **GPL**

MITM - MAN-IN-THE-MIDDLE

ATTAQUE | FR : HDM - ATTAQUE DE L'HOMME DU MILIEU

Attaque dans laquelle un acteur malveillant se place clandestinement entre deux parties qui communiquent, afin d'intercepter et potentiellement de modifier les messages échangés, sans que les deux parties remarquent sa présence.

Termes associés : **DIFFIE-HELLMAN** | **ECLIPSE**

MIXAGE

CONFIDENTIALITÉ | EN : MIXING

Dans le domaine général des mathématiques, le mixage ou le mélange se réfère à la propriété d'un système dynamique où, après un certain temps, toutes les portions de l'espace initial peuvent en théorie se retrouver mêlées avec n'importe quelle autre portion. Le mixage implique que la position d'une particule ou l'état d'un système évolue de telle manière que sa distribution future soit indépendante de sa distribution initiale, atteignant ainsi un état où les caractéristiques de l'état initial sont uniformément distribuées dans tout l'espace du système. Dans le cadre de Bitcoin, on peut utiliser cette notion pour évaluer la qualité d'un processus de mélange de pièces comme un coinjoin.

Certains bitcoiners différencient la notion de mixage du processus de coinjoin. En effet, ils disent que le mixage se réfère au mélange de pièces effectué par une entité possédant les fonds, contrairement aux coinjoins où l'utilisateur conserve toujours la possession des fonds. Toutefois, selon moi, cette distinction est incorrecte, car le coinjoin implique nécessairement un mixage au sens mathématique du terme.

Terme associé : **COINJOIN**

MODÈLE DE SCRIPT

SCRIPT | EN : SCRIPT TEMPLATE

Template permettant l'utilisation de scripts standards. Un modèle de script est essentiellement une petite liste d'*opcodes* mis ensemble pour former une norme qui spécifie une manière d'établir des conditions de dépenses sur des bitcoins. Voici quelques exemples de modèles de script : P2PK, P2PKH, P2WPKH, P2SH...

Termes associés : **P2PKH** | **SCRIPT**

MODÈLE DE TRANSACTION

PROTOCOLE | EN : TRANSACTION PATTERN

Un *pattern* de transaction est simplement un modèle ou une structure globale de transaction typique, que l'on peut retrouver sur la blockchain, et dont on connaît l'interprétation vraisemblable qui nous sera utile dans le cadre d'une analyse de chaîne. Lorsque l'on étudie les *patterns*, on va s'attarder sur une seule transaction que l'on va analyser à un niveau élevé (contrairement aux heuristiques internes et externes d'analyse de chaîne). En d'autres termes, nous allons uniquement regarder le nombre d'UTXOs en inputs et le nombre d'UTXOs en outputs, sans nous attarder sur les détails plus spécifiques ou l'environnement de la transaction. À partir du modèle observé, nous pourrons interpréter la nature de la transaction. On va alors rechercher des caractéristiques sur sa structure et en déduire une interprétation vraisemblable.

En anglais, on parle de « patterns ».

Termes associés : **ANALYSE DE CHAINE** | **PAIEMENT SIMPLE**

MODÈLE TEMPOREL

CONFIDENTIALITÉ | EN : TEMPORAL TEMPLATE

Certains comportements humains sont reconnaissables on-chain. Celui qui est le plus utile dans une analyse de chaîne, c'est peut-être votre rythme de sommeil ! Et oui, lorsque vous dormez, a priori, vous ne diffusez pas de transactions Bitcoin. Or, vous dormez généralement à peu près aux mêmes horaires. Il est donc courant d'utiliser des analyses temporelles dans l'analyse de chaîne. Il s'agit tout simplement du recensement des heures auxquelles les transactions d'une entité donnée sont diffusées au réseau Bitcoin. L'analyse de ces modèles temporels nous permet de déduire de nombreuses informations.

Tout d'abord, une analyse temporelle permet parfois d'identifier la nature de l'entité tracée. Si l'on observe que les transactions sont diffusées de manière constante sur 24 heures, alors cela va trahir une forte activité économique. L'entité derrière ces transactions est vraisemblablement une entreprise, potentiellement internationale et peut-être avec des procédures automatisées en interne. Au contraire, si l'on voit que le *pattern* temporel est plutôt réparti sur 16 heures bien spécifiques, alors on peut estimer que l'on a affaire à un utilisateur individuel, ou peut-être à une entreprise locale en fonction des volumes échangés.

Au-delà de la nature de l'entité observée, le *pattern* temporel peut également nous indiquer approximativement la localisation de l'utilisateur grâce aux fu-

seaux horaires. On pourra ainsi rapprocher d'autres transactions, et utiliser l'horodatage de celles-ci comme une heuristique supplémentaire pouvant s'ajouter dans une analyse de chaîne.

Dans un registre différent, c'est également une analyse temporelle de ce type qui a permis de formuler l'hypothèse selon laquelle Satoshi Nakamoto n'opérait pas depuis le Japon, mais bien depuis les États-Unis : *The Time Zones of Satoshi Nakamoto*.

Termes associés : **ANALYSE DE CHAINE** | **HEURISTIQUE D'ANALYSE**

MOSTRO

✱ OUTIL

Protocole d'échange pair-à-pair de bitcoins sans KYC, construit sur Nostr. Mostro permet d'acheter et de vendre des bitcoins contre des monnaies fiat via le Lightning Network. Le protocole utilise des *hold invoices* Lightning comme mécanisme de séquestre : les bitcoins du vendeur sont verrouillés dans une facture en attente et ne sont libérés vers l'acheteur qu'après confirmation de la réception du paiement fiat. L'utilisation de Nostr comme couche de communication rend le système résistant à la censure, contrairement aux solutions centralisées. Mostro est écrit en Rust et open source.

Termes associés : **KYC - KNOW YOUR CUSTOMER** | **LIGHTNING NETWORK** | **NOSTR**

MPP - MULTI-PATH PAYMENTS

ϟ LIGHTNING NETWORK | FR : PAIEMENTS MULTICHEMINS

Terme générique pour désigner toutes les techniques de paiement sur Lightning qui permettent de fractionner une transaction en plusieurs petites parties pour les acheminer via différentes routes. Autrement dit, chaque fraction de paiement emprunte un chemin de nœuds différent. Cela permet de contourner les limitations de liquidité sur un canal unique dans la route.

Les paiements multi-path offrent également de légers avantages en termes de confidentialité, car il devient plus difficile pour un observateur de lier chaque fraction de paiement à l'ensemble de la transaction. Toutefois, dans sa version de base, tous les fragments partagent le même secret pour les HTLCs, ce qui peut rendre la transaction traçable si un observateur est présent sur plusieurs routes. De plus, du fait que le secret est unique pour toutes les fractions du paiement, celui-ci n'est pas atomique. Cela signifie que certaines parties du paiement peuvent être exécutées avec succès, tandis que d'autres peuvent échouer. Ces inconvénients sont corrigés avec les « *Atomic Multi-Path Payment* ».

On parle également parfois de « Multi-Part Payment » pour désigner cette même méthode.

Terme associé : **ATOMIC MULTI-PATH PAYMENTS**

MS-SMT

</> INFORMATIQUE

Le *Merkle Sum Sparse Merkle Tree* (MS-SMT) est une structure de données combinant les propriétés d'un *Sparse Merkle Tree* (SMT) et d'un *Merkle Sum Tree*. Il permet à la fois de produire des preuves d'inclusion et de non-inclusion (grâce à sa nature « creuse » où chaque feuille a une position déterminée par le hash de sa clé), et de vérifier la conservation des sommes (grâce à la propagation des valeurs dans chaque nœud jusqu'à la racine). Dans le protocole Taproot Assets, le MS-SMT sert à engager les actifs dans la blockchain Bitcoin : sa racine est intégrée dans un tapscript d'une transaction Taproot. Cette structure garantit qu'il est possible de prouver qu'un actif existe, qu'un actif n'existe pas, et que le total des actifs émis est cohérent, empêchant toute inflation non autorisée.

Termes associés : **MERKLE SUM TREE** **SMT - SPARSE MERKLE TREE**

MTGOX

HISTOIRE

Plateforme d'échange de bitcoins fondée en 2010 par Jed McCaleb à Tokyo, au Japon, puis rachetée par Mark Karpelès en 2011. À son apogée, Mt. Gox traitait environ 70 % des transactions de bitcoins dans le monde, ce qui en faisait de loin la plus grande plateforme d'échange. En février 2014, Mt. Gox a suspendu les retraits puis a déclaré faillite après avoir révélé la perte d'environ 850 000 bitcoins, dont 750 000 appartenant à ses clients. Le nom « Mt. Gox » est un acronyme de « Magic : The Gathering Online eXchange », car le site avait été initialement conçu comme une place de marché pour des cartes de ce jeu. Cet effondrement reste l'un des événements les plus marquants de l'histoire de Bitcoin et a mis en lumière les risques liés à la conservation de ses bitcoins sur une plateforme centralisée.

Terme associé : **HALVING**

MTP - MEDIAN TIME PAST

PROTOCOLE | FR : TEMPS MÉDIAN PASSÉ

Concept utilisé dans le protocole Bitcoin pour déterminer une marge sur l'horodatage consensuel du réseau. Le MTP est défini comme la médiane des horodatages des 11 derniers blocs minés. L'utilisation de cet indicateur permet d'éviter les désaccords entre les nœuds sur l'heure exacte en cas de décalage. Le MTP était initialement utilisé pour vérifier la validité de l'horodatage des blocs par rapport au passé. Depuis le BIP-0113, il est également utilisé comme référentiel du temps du réseau pour vérifier la validité des opérations de verrouillages temporels (`nLockTime`, `OP_CHECKLOCKTIMEVERIFY`, `nSequence` et `OP_CHECKSEQUENCEVERIFY`).

Termes associés : **NLOCKTIME** **TIMELOCK**

MULTI PROTOCOL COMMITMENT

RGB

Dans le cadre du protocole RGB, le MPC désigne la structure d'arbre de Merkle conçue pour intégrer, au sein d'une unique transaction Bitcoin, plusieurs *Transition Bundles* issus de contrats distincts. Ce mécanisme permet de regrouper divers engagements, lesquels correspondent à des contrats ou des actifs différents, en un seul point d'ancrage. En consolidant ces transitions dans une structure hiérarchisée, le MPC optimise l'occupation de l'espace de bloc, afin de réduire les frais de transaction et d'améliorer l'efficacité de l'utilisation de la blockchain.

Termes associés : **ANCHOR** | **TRANSITION BUNDLE**

MULTISIG

SCRIPT

Les portefeuilles multisignatures, souvent abrégés « multisig », sont conçus pour renforcer la sécurisation de bitcoins en exigeant plusieurs signatures provenant de différentes clés privées pour autoriser une dépense. Cette méthode répartit le risque entre plusieurs clés, ce qui permet de réduire à la fois le risque de perte et celui de vol (selon la configuration du multisig). Les portefeuilles multisig fonctionnent selon un modèle « m-de-n », où `m` désigne le nombre minimal de signatures requises pour valider une transaction, et `n` le nombre total de clés impliquées. Par exemple, une configuration 2-de-3 nécessite deux signatures sur trois possibles pour valider une transaction. Cette approche offre une sécurité supérieure par rapport aux portefeuilles à clé unique, mais elle introduit également plus de complexité en termes de gestion et de sauvegarde. De plus, les transactions utilisant les anciens standards de multisig sont moins confidentielles et plus coûteuses en frais que les transactions *singlesig* classiques. Cependant, des innovations récentes telles que Taproot et l'utilisation de *descriptors* vont permettre de minimiser, voire d'éliminer, ces inconvénients des multisigs.

Certains bitcoiners distinguent les termes « Multisig » et « Multisig à seuil ». En effet, certains affirment qu'un multisig est forcément un n-de-n, tandis qu'un multisig à seuil est un m-de-n. Toutefois, dans le langage courant, il est accepté de parler de « Multisig » même pour m-de-n.

Termes associés : **BARE-MULTISIG** | **P2SH**

MUSIG

CRYPTOGRAPHIE

Protocole de multisignature Schnorr n-de-n proposé en 2018 par Gregory Maxwell, Andrew Poelstra, Yannick Seurin et Pieter Wuille. Il agrège les clés publiques de plusieurs signataires en une seule clé et produit une unique signature valide, indiscernable d'une signature individuelle sur la blockchain. Pour contrer l'attaque par clé frauduleuse (« *rogue key attack* »), dans laquelle un participant manipule sa clé publique pour contrôler seul la clé agrégée, le protocole multiplie chaque clé par un coefficient dérivé de l'ensemble des clés.

Le processus de signature se déroule en trois tours de communication : échange d'engagements cryptographiques sur les nonces, révélation des nonces, puis échange des signatures partielles. Cette structure en trois tours protège contre les attaques par manipulation de nonces, mais impose une interactivité contraignante. MuSig a été supplanté par MuSig2, qui ramène le processus à deux tours.

Termes associés : **MUSIG-DN** **MUSIG2** **SCHNORR**

MUSIG-DN

CRYPTOGRAPHIE

Variante de MuSig proposée en 2020 par Jonas Nick, Tim Ruffing, Yannick Seurin et Pieter Wuille. Le sigle « DN » signifie « Deterministic Nonces ». Le protocole élimine le tour d'engagement sur les nonces en imposant à chaque signataire de dériver ses nonces de manière déterministe à partir de sa clé privée et du message, puis de fournir une preuve à divulgation nulle de connaissance attestant la correction du calcul. Ce mécanisme ramène la signature à deux tours de communication tout en empêchant la manipulation des nonces.

Malgré son élégance théorique, la génération des preuves s'avère trop coûteuse en ressources pour un usage pratique, notamment sur les *hardware wallets*. MuSig-DN est resté une étape de recherche intermédiaire et a été supplanté par MuSig2, qui atteint le même objectif de deux tours avec une approche plus légère fondée sur des nonces aléatoires multiples.

Termes associés : **MUSIG** **MUSIG2**

MUSIG2

CRYPTOGRAPHIE

Évolution de MuSig qui réduit de trois à deux le nombre de tours nécessaires pour produire une multisignature Schnorr agrégée. Proposé en 2020 par Jonas Nick, Tim Ruffing et Yannick Seurin, le protocole élimine le tour d'engagement sur les nonces grâce à l'utilisation de plusieurs nonces aléatoires par signataire : chaque participant envoie ses nonces dès le premier tour, avant même de connaître le message à signer, puis les signatures partielles sont échangées au second tour.

Comme MuSig, la signature produite est indiscernable d'une signature individuelle sur la blockchain. La réutilisation de nonces entre sessions de signature compromet toutefois irrémédiablement la sécurité du schéma. MuSig2 a été

formalisé dans le BIP-0327.

Termes associés : **MUSIG** | **MUSIG-DN** | **SCHNORR**

MYNODE

OUTIL

Logiciel open source qui permet de déployer facilement un nœud complet Bitcoin et Lightning. MyNode propose une interface web intuitive pour gérer Bitcoin Core, LND et un ensemble d'applications complémentaires : explorateur de blocs, serveur Electrum, Mempool.space, BTCPay Server, et d'autres outils. MyNode existe en version gratuite (Community Edition) et en version payante (Premium) avec des fonctionnalités supplémentaires et un support dédié.

Termes associés : **LIGHTNING NETWORK** | **NOEUD**

MYSTERY SHOPPER PAYMENTS

ATTAQUE | FR : PAIEMENTS DE CLIENT MYSTÈRE

Attaque dans laquelle un attaquant envoie un petit paiement légitime en bitcoins à sa cible pour obtenir des informations et tenter de compromettre sa confidentialité. Cette technique est souvent utilisée contre les commerçants. Prenons l'exemple d'Alice, qui possède une boulangerie qui accepte les paiements en bitcoins. Eve, l'attaquante, pourrait acheter un croissant pour obtenir une adresse de réception d'Alice. Eve pourrait ensuite suivre les mouvements de cet UTXO pour tenter de déduire des informations supplémentaires sur Alice en utilisant des heuristiques de traçage.

Termes associés : **ANALYSE DE CHAINE** | **DUST**

N

NAKAMOTO SATOSHI

HISTOIRE

Pseudonyme de la personne ou du groupe qui a créé Bitcoin et écrit son livre blanc original en 2008 (White Paper). Nakamoto, qui a communiqué uniquement en ligne, a finalement disparu de la scène publique en 2011.

Termes associés : **CYPHERPUNKS** **WHITE PAPER**

NAMECOIN

HISTOIRE

Premier altcoin de l'histoire de Bitcoin lancé en avril 2011. Namecoin est un fork du code source de Bitcoin qui vise à fournir un système décentralisé d'enregistrement et de gestion de noms de domaine, en alternative au DNS (*Domain Name System*) traditionnel. Son principal cas d'usage est l'extension `.bit`, qui permet d'enregistrer des noms de domaine résistants à la censure directement sur la blockchain.

Termes associés : **ALTCOIN** **MINAGE FUSIONNÉ**

NAT - NETWORK-ADJUSTED TIME

PROTOCOLE | FR : TEMPS AJUSTÉ PAR LE RÉSEAU

Estimation du temps universel établie sur les horloges des nœuds du réseau. Chaque nœud calcule son NAT en prenant la médiane des différences de temps entre son horloge locale (UTC) et celles des nœuds avec lesquels il est connecté, puis en additionnant son horloge locale avec la médiane de ces différences, jusqu'à un maximum de 70 minutes. Le *network-adjusted time* est donc une médiane du temps des nœuds calculée en local par chaque nœud. Ce référentiel est ensuite utilisé pour vérifier la validité des horodatages des blocs. En effet, pour qu'un bloc soit accepté par un nœud, son horodatage doit se situer entre le MTP (temps médian des 11 derniers blocs minés) et le NAT plus 2 heures :

```
MTP < Horodatage valide < (NAT + 2h)
```

Termes associés : **ENTÊTE DE BLOC** **MTP - MEDIAN TIME PAST**

NBITS

PROTOCOLE

Champ de 4 octets présent dans l'en-tête de chaque bloc Bitcoin, qui encode la cible de difficulté sous une forme compacte : le premier octet indique le nombre d'octets de la cible complète (l'exposant), et les trois octets suivants constituent la mantisse (les chiffres significatifs de la cible). La formule de décodage est : $cible = mantisse \times 256^{(exposant-3)}$.

Un hash de bloc doit être inférieur ou égal à la valeur cible encodée dans `nBits` pour que le bloc soit considéré comme valide. Ce champ est recalculé tous les 2 016 blocs lors de l'ajustement de la difficulté, afin de maintenir un intervalle moyen d'environ 10 minutes entre chaque bloc.

Termes associés : **CIBLE DE DIFFICULTÉ** **ENTÊTE DE BLOC**

NERDMINER

↗ MINAGE

Projet *open source* de minage solo de Bitcoin qui fonctionne sur des microcontrôleurs ESP32. Créé par BitMaker en 2022, il permet d'assembler un petit mineur à très faible consommation pour découvrir concrètement le fonctionnement du minage. L'appareil peut être connecté à une pool de minage solo comme Public Pool. Avec un *hashrate* d'environ 78 kH/s, les chances de trouver un bloc sont astronomiquement faibles, mais le projet a avant tout une vocation éducative et ludique. NerdMiner affiche sur son écran intégré des statistiques en temps réel sur le hachage et le réseau Bitcoin.

Termes associés : **ASIC** | **MINAGE**

NESTED SEGWIT

✪ PROTOCOLE | FR : SEGWIT IMBRIQUÉ

Standard de scripts utilisé pour envelopper des scripts SegWit natifs, au sein d'un script P2SH. Les scripts Nested SegWit ont été inventés au lancement de SegWit pour faciliter son adoption. Ils permettent d'utiliser ce nouveau standard, même avec des services ou des *wallets* pas encore compatibles nativement avec SegWit. C'est une sorte de script de transition vers la nouvelle norme. Aujourd'hui, il n'est donc plus très pertinent d'utiliser ce type de scripts SegWit *wrappés*, puisque la plupart des *wallets* ont implémenté du SegWit natif.

Terme associé : **P2SH-P2WPKH**

NEUTRINO

✱ OUTIL

Client léger Bitcoin écrit en Go et développé par Lightning Labs. Conçu pour les applications mobiles utilisant le Lightning Network, Neutrino implémente le protocole de *compact block filters* défini dans les BIP-0157 et BIP-0158 pour synchroniser la blockchain en minimisant la bande passante et le stockage.

Neutrino ne télécharge que les en-têtes de blocs et les en-têtes de filtres compacts. Les filtres et les blocs complets sont chargés à la demande lorsqu'une correspondance est détectée avec les scripts surveillés. Il propose des fonctionnalités de *rescan* pour analyser la blockchain à la recherche de transactions pertinentes, ainsi qu'une fonction `GetUtxo` pour vérifier l'existence et l'état de dépense d'une sortie.

Neutrino est notamment utilisé comme *backend* par LND pour les configurations légères ne nécessitant pas un nœud Bitcoin Core complet. Le terme « Neutrino » est parfois utilisé par abus de langage pour désigner le protocole BIP-0157/0158 lui-même, alors qu'il ne désigne que cette bibliothèque cliente spécifique.

Termes associés : **COMPACT BLOCK FILTERS** | **LND**

NFT

COUCHE SUPÉRIEURE

Acronyme de *Non-Fungible Token*. Un NFT représente une unité de bitcoin qui, par des utilisateurs d'un protocole externe à Bitcoin, est interprétée non fongible et associée à la propriété d'un actif externe à Bitcoin, tel qu'une image ou tout autre type de donnée.

Sur Bitcoin, des protocoles comme Ordinals et Stamps permettent la création et la gestion de ces NFTs. Ces protocoles sont souvent controversés, car leur activité encombre la blockchain avec des données non financières. Cela peut avoir des conséquences négatives sur le système, notamment induire une charge opérationnelle accrue pour les nœuds complets ou augmenter les frais de transaction par l'influence de facteurs économiques externes.

Termes associés : **INSCRIPTIONS** **ORDINALS THEORY**

NIP

RÉSEAU

Spécification technique qui définit une fonctionnalité, un format ou un comportement attendu dans le protocole Nostr. Les NIPs jouent un rôle analogue aux BIPs de Bitcoin : ce sont des documents communautaires qui décrivent de manière formelle comment les clients et les relais doivent implémenter une fonctionnalité donnée. Chaque NIP porte un numéro et couvre un aspect précis du protocole. À l'exception du NIP-01 qui décrit le protocole de base que tout le monde devrait implémenter, les NIPs sont optionnels et rétrocompatibles. Un NIP peut être marqué comme `unrecommended` lorsqu'il est déprécié ou remplacé par une spécification plus récente. Le dépôt officiel des NIPs est maintenu sur GitHub et ouvert aux contributions.

Termes associés : **BIP** **NOSTR**

NLOCKTIME

SCRIPT

Champ intégré dans les transactions qui définit une condition temporelle avant laquelle la transaction ne peut être ajoutée à un bloc valide. Ce paramètre permet de spécifier un temps précis (timestamp Unix) ou un nombre de blocs spécifique comme condition pour que la transaction soit considérée comme valide. C'est donc un timelock absolu (pas relatif). Le `nLockTime` agit sur l'intégralité de la transaction et permet effectivement de vérifier le temps, alors que l'opcode `OP_CHECKLOCKTIMEVERIFY` permet uniquement de comparer la valeur en haut de la pile avec la valeur du `nLockTime`.

Terme associé : **TIMELOCK**

NO2X

HISTOIRE

Nom de la campagne lancée en août 2017 suite au verrouillage du soft fork SegWit dont le but est de convaincre les utilisateurs de ne pas implémenter le doublement de la taille des blocs de 1 Mo à 2 Mo tel que prévu dans la mise à jour SEGWIT2X.

Termes associés : **BLOCKSIZE WAR** **SEGWIT2X**

NODL

ORGANISATION

Entreprise fondée fin 2018 qui conçoit des nœuds Bitcoin et Lightning Network clé en main, sous forme de boîtiers matériels prêts à l'emploi. Nodl a été le premier fabricant à intégrer BTCPay Server, à imposer Tor par défaut sur tous ses appareils, et à proposer le chiffrement complet du disque avec un interrupteur physique de sécurité (*kill switch*). L'entreprise propose également nodl Cloud, un service d'infrastructure hébergée dans ses propres centres de données (Paris, Genève, Zurich) pour les particuliers et entreprises souhaitant faire tourner des nœuds sans gérer le matériel.

Termes associés : **NOEUD COMPLET** **NOEUD LIGHTNING**

NOEUD

PROTOCOLE | EN : NODE

Dans le réseau Bitcoin, un nœud est un ordinateur qui exécute un client du protocole Bitcoin (comme Bitcoin Core par exemple). Il participe au réseau en maintenant une copie de la blockchain, en relayant et en vérifiant les transactions et les nouveaux blocs et, optionnellement, en participant au processus de minage. La somme de tous les nœuds Bitcoin représente le réseau Bitcoin en lui-même.

Il existe plusieurs types de nœuds sur Bitcoin, dont les nœuds complets et les nœuds légers. Les nœuds complets conservent une copie intégrale de la blockchain, vérifient toutes les transactions et les blocs selon les règles de consensus, et participent activement à la diffusion de transactions et de blocs sur le réseau. En revanche, les nœuds légers, ou nœuds SPV (*Simple Payment Verification*), ne conservent que les entêtes des blocs et comptent sur les nœuds complets pour obtenir des informations sur les transactions.

Certains différencient également les nœuds dits « élagués » (« pruned node *» en anglais). Ce sont des nœuds complets, qui téléchargent et vérifient tous les blocs depuis le bloc de Genèse, mais qui ne conservent que les blocs les plus récents en mémoire.*

Termes associés : **NOEUD COMPLET** **NOEUD SPV - NOEUD LÉGER**

NOEUD COMPLET

PROTOCOLE | EN : FULL NODE

Ordinateur qui exécute un client du protocole Bitcoin, et qui télécharge, vérifie et stocke la totalité de la blockchain, soit l'historique complet des transactions depuis le bloc de Genèse. Un nœud complet vérifie de manière autonome toutes les transactions et les blocs en fonction des règles de consensus de Bitcoin. C'est donc ce type de nœud qui procure le plus haut niveau de vérification pour son utilisateur, et qui permet de réduire au maximum le besoin de confiance envers une tierce partie. Le nœud complet nécessite plus de ressources de stockage, de puissance de calcul, de RAM et de bande passante qu'un nœud léger (SPV).

Termes associés : **NOEUD** | **NOEUD ÉLAGUÉ**

NŒUD DE ROUTAGE

LIGHTNING NETWORK | EN : ROUTING NODE

Nœud du Lightning Network qui achemine les paiements entre d'autres nœuds en servant d'intermédiaire dans le routage des transactions. Lorsqu'un paiement est envoyé entre deux utilisateurs qui ne partagent pas de canal direct, le réseau Lightning identifie un chemin composé d'un ou plusieurs nœuds de routage pour relayer les fonds de proche en proche via des HTLC. Chaque nœud de routage verrouille temporairement les fonds dans un canal, puis les transmet au nœud suivant, en percevant éventuellement de petits frais de routage en contrepartie de ce service. Les nœuds de routage jouent un rôle essentiel pour la connectivité du réseau Lightning, car ils permettent d'atteindre n'importe quel destinataire sans nécessiter de canal direct avec chaque contrepartie. Un bon nœud de routage maintient une liquidité suffisante et une bonne connectivité avec les autres nœuds du réseau.

Par extension, opérer un nœud de routage peut constituer une stratégie pour un opérateur de nœud souhaitant générer des revenus grâce aux frais de routage.

Termes associés : **HTLC** | **LIGHTNING NETWORK**

NOEUD ÉLAGUÉ

PROTOCOLE | EN : PRUNED NODE

Nœud complet qui exécute un élagage de la blockchain. Cela consiste à supprimer de manière progressive les blocs les plus anciens, après les avoir dûment vérifiés, pour conserver seulement les blocs les plus récents. La limite de conservation est renseignée dans le fichier `bitcoin.conf` via le paramètre `prune=n`, où `n` est la taille maximale prise par les blocs en mégaoctets (Mo). Si `0` est noté après ce paramètre, alors l'élagage est désactivé, et le nœud conserve la blockchain dans son intégralité.

Les nœuds élagués sont parfois considérés comme des types de nœuds différents des nœuds complets. L'utilisation d'un nœud élagué peut s'avérer particulièrement intéressante pour les utilisateurs confrontés à des contraintes en termes de capacité de stockage. Actuellement, un nœud complet doit disposer de plus de 800 Go rien que pour le stockage de la blockchain. Un nœud élagué

peut limiter le stockage requis jusqu'à 550 Mo. Bien qu'il utilise moins d'espace disque, un nœud élagué maintient un niveau de vérification et de validation semblable à celui d'un nœud complet. Les nœuds élagués offrent donc plus de confiance à leurs utilisateurs en comparaison avec les nœuds légers (SPV).

Termes associés : **BITCOIN.CONF** | **NOEUD COMPLET**

NOEUD LIGHTNING

LIGHTNING NETWORK | EN : LIGHTNING NODE

Ordinateur qui exécute une implémentation du Lightning Network (Eclair, LND, Core Lightning...). Un nœud Lightning est établi sur un nœud Bitcoin, et permet de créer et de gérer des canaux de paiement bidirectionnels entre différents utilisateurs. Les nœuds Lightning sont connectés les uns aux autres, ce qui forme un réseau de canaux de paiement. Ces canaux permettent de réaliser un nombre presque illimité de transactions sans avoir besoin de les publier individuellement sur la blockchain Bitcoin.

Termes associés : **CANAL DE PAIMENT** | **LIGHTNING NETWORK**

NOEUD SPV - NOEUD LÉGER

PROTOCOLE | EN : SPV NODE - LIGHTWEIGHT NODE

Un nœud SPV (*Simplified Payment Verification*), parfois nommé « nœud léger », est un client léger d'un nœud Bitcoin qui permet aux utilisateurs de valider les transactions sans avoir à stocker l'intégralité de la blockchain. Au lieu de cela, un nœud SPV stocke seulement les entêtes des blocs, et obtient des informations sur des transactions spécifiques en interrogeant des nœuds complets lorsque nécessaire. Ce principe de vérification est rendu possible par la structure des transactions dans les blocs Bitcoin, qui sont organisées au sein d'un accumulateur cryptographique (Arbre de Merkle).

Cette approche est avantageuse pour les appareils avec des ressources limitées, tels que les téléphones portables. Cependant, les nœuds SPV font confiance aux nœuds complets pour la disponibilité des informations, ce qui implique un niveau de confiance supplémentaire et, par conséquent, une moindre sécurité par rapport aux nœuds complets. Les nœuds SPV ne peuvent pas valider les transactions de manière autonome, mais ils peuvent vérifier si une transaction est incluse dans un bloc en consultant les preuves de Merkle.

Termes associés : **ARBRE DE MERKLE** | **SIMPLIFIED PAYMENT VERIFICATION**

NOISE

RÉSEAU

Framework cryptographique utilisé pour établir et authentifier les connexions entre nœuds Lightning. Le protocole Noise, spécifié dans le BOLT-08, définit le *handshake* initial lors duquel deux nœuds échangent leurs clés et négocient une session chiffrée. Il repose sur un échange de clés Diffie-Hellman avec des clés éphémères et statiques, pour avoir des propriétés de confidentialité persistante (*forward secrecy*). Une fois le *handshake* terminé, toutes les communications entre les deux nœuds sont chiffrées et authentifiées. Noise remplace les protocoles TLS ou SSH habituellement utilisés dans les applications réseau, avec une spécification plus simple et mieux adaptée aux contraintes du Lightning Network.

Termes associés : **LIGHTNING NETWORK** | **TOR**

NONCE

CRYPTOGRAPHIE

Dans le contexte de l'informatique, le terme « nonce » désigne un nombre utilisé seulement une seule fois, puis remplacé. Il est souvent aléatoire ou pseudo-aléatoire. On l'utilise dans divers protocoles cryptographiques pour garantir la sécurité du procédé. Par exemple, les signatures ECDSA utilisées au sein du protocole Bitcoin incluent l'utilisation d'un nonce. Cela veut dire que ce nombre doit être nouveau pour chaque signature. Si ce n'est pas le cas, il est possible de calculer la clé privée utilisée en rapprochant deux signatures qui utilisent le même nonce.

On utilise également des nonces dans le processus de minage sur Bitcoin. Les mineurs incrémentent ces valeurs modifiables au sein de leurs blocs candidats. Ils modifient la valeur du nonce dans le but de trouver une empreinte cryptographique inférieure ou égale à la cible de difficulté. Ce processus nécessite une grande puissance de calcul, car il s'agit d'une recherche exhaustive parmi un grand nombre de nonces possibles. Lorsqu'un mineur trouve un nonce qui, lorsqu'il est inclus dans son bloc, produit un hash répondant aux critères de difficulté, le bloc est diffusé au réseau, et le mineur remporte la récompense.

En 2010, des chercheurs ont découvert que la PlayStation 3 de Sony utilisait le même nonce lors de la signature de différents paquets de code. Cette réutilisation du nonce a permis aux attaquants de calculer la clé privée utilisée pour signer le logiciel. Avec la clé privée en main, les attaquants pouvaient créer des signatures valides pour n'importe quel code, ce qui leur permettait d'exécuter des logiciels non autorisés, y compris des jeux piratés ou des systèmes d'exploitation personnalisés, directement sur la PS3.

Il existe une idée reçue sur l'origine du terme de « nonce ». Certains affirment qu'il représenterait l'abréviation de « number only used once ». En réalité, l'origine du mot remonte au 12ème siècle et provient de l'évolution sémantique de l'expression en moyen anglais « for then anes » qui signifiait « pour cette occasion ».

Termes associés : **ECDSA** | **MINAGE**

NOSTR

RÉSEAU

Protocole ouvert et décentralisé de communication, dont l'acronyme signifie *Notes and Other Stuff Transmitted by Relays*. Créé par fiatjaf en 2020, Nostr repose sur une architecture simple : des clients signent cryptographiquement des événements (la structure de données unique du protocole) et les publient vers des relais, des serveurs qui stockent et redistribuent ces événements. L'identité d'un utilisateur est une paire de clés cryptographiques sur la courbe secp256k1, la même que celle utilisée par Bitcoin. Aucun serveur central ne contrôle l'accès ou le contenu : si un relais censure un utilisateur, celui-ci peut migrer vers d'autres relais sans perdre son identité ni son audience. Nostr est étroitement lié à l'écosystème Bitcoin, notamment via l'intégration native des *zaps* (micropaiements Lightning) et l'utilisation de la même infrastructure cryptographique. Le protocole est extensible grâce aux NIPs (*Nostr Implementation Possibilities*), des spécifications communautaires qui définissent les formats d'événements et les fonctionnalités. Au-delà du *micro-blogging*, Nostr sert de socle à des applications variées : messagerie chiffrée, articles longs, *marketplaces*, groupes fermés ou encore collaboration de code via Git.

Termes associés : LIGHTNING NETWORK | SECP256K1

NPUB

RÉSEAU

Format d'encodage de la clé publique Nostr en Bech32, défini par le NIP-19. Le préfixe `npub` identifie une clé publique Nostr. La clé publique brute est un nombre de 32 octets sur la courbe secp256k1, dérivé depuis sa clé privée. L'encodage npub ajoute un préfixe lisible par l'humain (`npub1…`), une somme de contrôle pour détecter les erreurs de saisie, et utilise un alphabet excluant les caractères ambigus. La npub constitue l'identifiant public d'un utilisateur Nostr : il peut être partagé librement et permet à quiconque de retrouver le profil et les publications associés. Contrairement au nsec, il ne compromet pas la sécurité du compte s'il est divulgué.

Termes associés : BECH32 ET BECH32M | NOSTR

NSEC

RÉSEAU

Encodage Bech32 (NIP-19) d'une clé secrète (privée) Nostr, avec le préfixe `nsec`. La clé correspond à une valeur de 32 octets utilisée comme secret sur secp256k1 ; elle permet de produire les signatures Schnorr des événements Nostr et donc de prouver le contrôle de l'identité (via la `npub`). La perte de la nsec entraîne la perte de contrôle de cette identité. La nsec ne doit jamais être partagée ni saisie dans un client web non fiable ; pour réduire l'exposition, le NIP-07 standardise `window.nostr`, via lequel une extension peut signer les événements sans divulguer la clé à l'application.

Termes associés : CLÉ PRIVÉE | NOSTR

NSEQUENCE

SCRIPT

Le champ `nSequence` dans une entrée de transaction Bitcoin est utilisé pour indiquer la manière dont cette entrée est verrouillée dans le temps. À l'origine, il visait à permettre le remplacement dynamique de transactions dans les mempools afin de permettre un système de paiement en surcouche similaire à Lightning. Toutefois, son utilisation a évolué avec l'introduction du timelock relatif via le BIP-0068. Le champ `nSequence` peut désormais spécifier un délai relatif avant qu'une transaction soit incluse dans un bloc. Ce délai peut être défini en termes de nombre de blocs, ou bien comme un multiple de 512 secondes (c'est-à-dire, du temps réel). Notons que cette nouvelle interprétation du champ `nSequence` est uniquement valide si le champ `nVersion` est supérieur ou égal à `2`. Cette interprétation du champ `nSequence` se fait au niveau des règles de consensus de Bitcoin. Par ailleurs, au niveau des règles de standardisation, ce champ est également utilisé pour le signalement de RBF. Si une transaction inclut un `nSequence` inférieur à `0×fffffffe`, alors elle pourra être remplacée via RBF sur les nœuds qui suivent cette politique.

Terme associé : **TIMELOCK**

NULL DATA

PROTOCOLE

Type de transaction Bitcoin qui permet d'insérer une petite quantité de données arbitraires grâce à un `OP_RETURN`. Les bitcoins éventuellement associés à ce type d'output sont non dépensables de manière prouvée, car l'`OP_RETURN` signale un script invalide.

Termes associés : **OP_RETURN - 0X6A** **OUTPUT**

NULLDUMMY

PROTOCOLE

Règle de consensus introduite avec le BIP-0147 dans le soft fork SegWit qui exige que l'élément factice (« *dummy element* ») utilisé dans les opcodes `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY` soit un tableau d'octets vide (`OP_0`). Cette mesure a été mise en place pour éliminer un vecteur de malléabilité en interdisant toute valeur autre que `OP_0` pour cet élément.

Termes associés : **BIP-0147** **DUMMY ELEMENT**

NUNCHUK

PORTEFEUILLE

Logiciel de portefeuille Bitcoin spécialisé dans le multisig, disponible sur ordinateur et mobile (iOS, Android). Nunchuk facilite la création et la gestion de portefeuilles à signatures multiples en proposant une interface accessible, même pour des utilisateurs non techniques. L'application prend en charge la plupart des *hardware wallets* du marché et utilise les PSBT pour coordonner les signatures entre plusieurs appareils. Nunchuk propose également des fonctionnalités collaboratives qui permettent à plusieurs personnes de cosigner des transactions de manière asynchrone. Le logiciel inclut des options de récupération assistée et d'héritage. Il existe en version gratuite avec les fonctionnalités de base et en version payante avec des services supplémentaires comme le multisig assisté.

Termes associés : **HARDWARE WALLET** | **MULTISIG**

NVERSION

PROTOCOLE

Le champ `nVersion` dans une transaction Bitcoin sert à indiquer la version du format de transaction utilisé. Il permet au réseau de distinguer les différentes évolutions du format de transaction au fil du temps, et d'appliquer les règles correspondantes. Ce champ a un impact au niveau des règles de consensus : lorsque `nVersion` $\geqslant$ `2`, l'interprétation du champ `nSequence` comme un timelock relatif est activée, conformément au BIP-0068. Toute valeur attribuée à ce champ n'entraîne cependant pas l'invalidation de la transaction. En revanche, le champ `nVersion` dispose de règles de standardisation qui n'acceptent que les valeurs de `1`, de `2` et de `3` actuellement. La version `3` a été introduite avec Bitcoin Core 28.0 pour les transactions TRUC (*Topologically Restricted Until Confirmation*), telles que définies dans le BIP-0431. Pour le moment, ce champ est principalement utile pour l'activation du champ `nSequence` et pour le signalement des transactions TRUC.

Terme associé : **NSEQUENCE**

NWC - NOSTR WALLET CONNECT

LIGHTNING NETWORK

Protocole défini par le NIP-47 qui permet de contrôler un portefeuille Lightning à distance en utilisant Nostr comme couche de communication. NWC établit une connexion chiffrée entre un client Nostr et un portefeuille Lightning via les relais Nostr, permettant au client d'initier des paiements, de générer des factures et de consulter le solde sans que le portefeuille soit intégré directement dans l'application. La connexion est authentifiée par une paire de clés Nostr dédiée, et les messages échangés entre le client et le portefeuille sont chiffrés de bout en bout. NWC est particulièrement utile pour séparer l'interface utilisateur de la gestion des fonds : un utilisateur peut par exemple connecter son nœud Lightning personnel à n'importe quel client Nostr compatible, ce qui facilite l'envoi de zaps ou les paiements Lightning en déplacement, sans exposer ses clés

privées à l'application.

Termes associés : **LIGHTNING NETWORK** | **NOSTR**

NYA - NEW YORK AGREEMENT

HISTOIRE | FR : ACCORD DE NEW YORK

Réunion privée qui s'est tenue en 2017, rassemblant plus de 50 entreprises de l'écosystème Bitcoin, à la suite de la conférence Consensus 2017. L'objectif de cette réunion était de finir le débat de longue date sur le passage à l'échelle de Bitcoin en parvenant à un accord. De cette rencontre est née la proposition SegWit2x, s'inspirant de la précédente proposition SegWit2Mb. Elle prévoyait deux modifications majeures du protocole Bitcoin :

- L'adoption de SegWit avec un seuil d'activation fixé à 80 pour cent de signalisations ;
- Un hard fork destiné à augmenter la taille maximale des blocs de 1 Mo à 2 Mo.

Malgré le signalement positif de plus de 80 % des mineurs, le projet n'a pas su rallier un consensus suffisant, aboutissant à son abandon. Cet événement a été interprété par de nombreux utilisateurs et développeurs comme une tentative d'attaque de Bitcoin.

Termes associés : **BLOCKSIZE WAR** | **SEGWIT2X**

O

OAP - OPEN ASSETS PROTOCOL

COUCHE SUPÉRIEURE

Le Protocole Open Assets (OAP), conçu par Flavien Charlon en 2013, représente la première mise en œuvre fonctionnelle des Colored Coins. Ce protocole permettait de stocker et de transférer des actifs non natifs sur la blockchain Bitcoin, sous la forme de tokens dénommés « Colored Coins ». Ces derniers sont marqués spécifiquement pour symboliser une promesse, qu'elle soit formelle ou informelle, d'échange contre des biens ou des services réels.

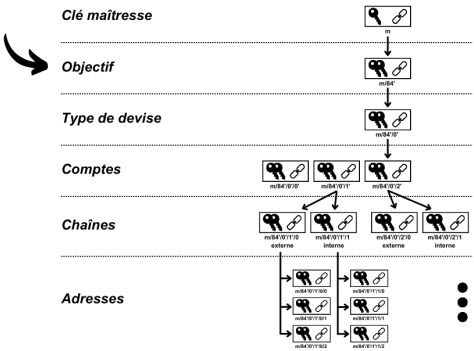
Termes associés : **COLORED COINS** | **OP_RETURN - 0X6A**

OBJECTIF

PORTEFEUILLE | EN : PURPOSE

Dans les portefeuilles déterministes et hiérarchiques (HD), l'objectif, défini par le BIP-0043, représente un niveau de dérivation spécifique. Cet index, situé à la première profondeur de l'arborescence de dérivation (`m / purpose' /`), identifie le standard de dérivation adopté par le portefeuille, afin de faciliter la compatibilité entre différents logiciels de gestion de portefeuille. Par exemple, dans le cas des adresses SegWit (BIP-0084), l'objectif est noté `/84'/`. Cette méthode permet d'organiser efficacement les clés entre les différents types d'adresses au sein d'un même portefeuille HD. Les index standards utilisés sont :

- Pour du P2PKH : `44'` ;
- Pour du P2WPKH-nested-in-P2SH : `49'` ;
- Pour du P2WPKH : `84'` ;
- Pour du P2TR : `86'`.



Terme associé : **CHEMIN DE DÉRIVATION**

OBOE - OFF-BY-ONE ERROR

SCRIPT | FR : ERREUR DE DÉCALAGE UNITAIRE

Erreur de logique où une boucle itère une fois de trop ou de moins, souvent due à une mauvaise utilisation des opérateurs de comparaison ou de mauvais indices dans la gestion des structures de données. Deux instances notables de ce bug existent dans Bitcoin.

La plus conséquente se trouve dans le calcul de l'ajustement de la difficulté. L'algorithme mesure le temps écoulé durant une période de 2016 blocs en comparant l'horodatage du premier bloc de la période à celui du dernier, puis compare cette durée au temps cible de $1\,209\,600$ secondes (2016×600). Le problème est que la différence entre le premier et le dernier bloc d'une période ne couvre que 2015 intervalles de blocs, et non 2016 :

$$T = t_{2015} - t_0$$

Les intervalles comptés sont :

$$\underbrace{[t_0 \to t_1]}_{1},\ \underbrace{[t_1 \to t_2]}_{2},\ \ldots,\ \underbrace{[t_{2014} \to t_{2015}]}_{2015}$$

Soit 2015 intervalles. Pour en obtenir 2016, il aurait fallu utiliser le dernier bloc de la période précédente comme point de départ :

$$T_{correct} \ = \ t_{2015} \ - \ t_{-1}$$

Où t_{-1} désigne l'horodatage du dernier bloc de la période précédente (bloc $n-1$ pour une période qui commence au bloc n). L'algorithme compare donc 2015 intervalles réels au temps théorique de 2016 intervalles. Ce décalage introduit un biais d'environ $\frac{1}{2016} \approx 0{,}05\,\%$ vers une difficulté légèrement surestimée.

Ce bug a surtout pour conséquence que les périodes de retarget ne se chevauchent pas : l'horodatage du dernier bloc d'une période n'entre pas dans le calcul de la période suivante. Cette discontinuité rend possible l'attaque *time warp*, où un attaquant majoritaire manipule l'horodatage du dernier bloc d'une période sans que cela n'affecte le calcul de la période suivante. Corriger réellement ce bug nécessiterait un hard fork, mais les attaques *time warp* peuvent être empêchées via un simple soft fork en ajoutant des contraintes sur les horodatages.

La seconde instance de bug *off-by-one* dans Bitcoin est le *dummy element* dans `OP_CHECKMULTISIG`, où l'opcode consomme un élément de pile supplémentaire par erreur. Ce bug a été corrigé dans les faits par le BIP-0147 (*NULLDUMMY*), qui impose que cet élément soit une valeur nulle.

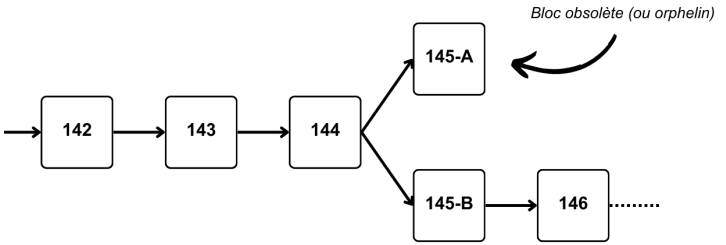
Termes associés : AJUSTEMENT DE LA DIFFICULTE | BIP-0147 | DUMMY ELEMENT | TIME WARP

OBSOLÈTE - BLOC

PROTOCOLE | EN : STALE / OBSOLETE - BLOCK

Fait référence à un bloc sans enfant : un bloc valide, mais exclu de la chaîne principale de Bitcoin. Il se produit lorsque deux mineurs trouvent un bloc valide sur une même hauteur de chaîne durant un court laps de temps et le diffusent sur le réseau. Les nœuds finissent par choisir un seul bloc à inclure dans la chaîne, selon le principe de la chaîne avec le plus de travail accumulé, rendant l'autre « obsolète ». Le processus menant à la production d'un bloc obsolète est le suivant :

- Deux mineurs trouvent un bloc valide à une même hauteur de chaîne durant un court intervalle de temps. Nommons-les `Bloc A` et `Bloc B` ;
- Chacun diffuse son bloc au réseau de nœuds Bitcoin. Chaque nœud dispose maintenant de 2 blocs à une même hauteur. Il existe donc deux chaînes valides ;
- Les mineurs continuent de chercher des preuves de travail pour les blocs suivants, mais pour ce faire, ils doivent obligatoirement choisir un seul bloc entre le `Bloc A` et le `Bloc B` au-dessus duquel ils vont miner ;
- Un mineur trouve finalement un bloc valide au-dessus du `Bloc B`. Appelons le `Bloc B+1` ;
- Il diffuse `Bloc B+1` aux nœuds du réseau ;
- Puisque les nœuds suivent la chaîne la plus longue (avec le plus de travail accumulé), ils vont estimer que la `Chaîne B` est celle qu'il faut suivre ;
- Ils vont abandonner le `Bloc A` qui ne fait plus partie de la chaîne principale. Il est donc devenu un bloc obsolète.



En français, on peut également dire « bloc périmé » ou « bloc abandonné ». Même si je ne suis pas en accord avec cet usage, certains bitcoiners utilisent le terme de « bloc orphelin » pour désigner ce qui est en réalité un bloc obsolète.

Terme associé : **ORPHELIN**

OCEAN

MINAGE

Pool de minage de Bitcoin lancée en novembre 2023 par Luke Dashjr, développeur historique de Bitcoin Core et Bitcoin Knots. Ocean se distingue par son approche axée sur la décentralisation du minage : les mineurs peuvent sélectionner ou construire eux-mêmes leurs propres *templates* de blocs, ce qui réduit la dépendance envers les opérateurs de pool centralisés. La pool utilise le sys-

tème de rémunération TIDES (*Transparent Index of Distinct Extended Shares*), qui récompense les mineurs directement via les sorties des transactions *coinbase*. Le projet s'inscrit dans la continuité d'Eligius, la pool de minage que Luke Dashjr avait lancée en 2011.

Termes associés : **MINAGE** | **MINING POOL**

OCTET

‹/› INFORMATIQUE | EN : BYTE

Unité de mesure de données informatiques équivalant à 8 bits. Chaque bit est un chiffre binaire (`0` ou `1`), ce qui signifie qu'un octet peut représenter 256 (2^8) combinaisons uniques.

Termes associés : **BINAIRE** | **BIT**

OFF-CHAIN

COUCHE SUPÉRIEURE | FR : HORS-CHAÎNE

Fait référence aux transactions ou activités plus ou moins liées à Bitcoin qui se produisent en dehors de la blockchain principale, mais qui disposent d'un lien ou d'un ancrage avec celle-ci. Elles ne sont pas immédiatement enregistrées sur la blockchain, mais nécessitent des mécanismes supplémentaires pour assurer leur sécurité et leur finalité. Ces opérations se justifient souvent par un désir d'outrepasser les limitations techniques inhérentes à Bitcoin, afin de disposer de transactions à finalité rapide, à bas frais, avec plus de capacité ou de fonctionnalités.

Termes associés : **LIGHTNING NETWORK** | **ON-CHAIN**

OFF-GRID

MINAGE | FR : HORS RÉSEAU

Désigne une opération de minage de Bitcoin qui fonctionne de manière autonome, sans être connectée au réseau électrique principal. Les mineurs off-grid utilisent généralement des sources d'énergie alternatives comme l'énergie solaire, éolienne, hydroélectrique, ou encore avec des générateurs fonctionnant aux combustibles fossiles (par exemple, le gaz destiné au torchage). Ce modèle est souvent privilégié pour réduire les coûts d'énergie, car il se base en partie sur de l'énergie qui, sans le minage, serait gaspillée. Les mineurs off-grid ont l'avantage de pouvoir s'installer dans des endroits isolés, afin de consommer cette énergie trop éloignée des populations.

Le minage off-grid présente souvent des avantages écologiques, car il peut, dans certains cas, aider à réduire les émissions de gaz à effet de serre en limitant le torchage. De plus, il rend économiquement viables certains projets de production d'énergie renouvelable malgré leur intermittence naturelle.

Termes associés : **BEHIND-THE-METER** | **ON-GRID**

OFFER

LIGHTNING NETWORK | FR : OFFRE / DEMANDE DE PAIEMENT RÉUTILISABLE

Dans le cadre de BOLT-12, les « *offers* » sont des QR codes statiques pour effectuer plusieurs paiements sur le Lightning Network. Contrairement aux invoices classiques, les *offers* peuvent être réutilisées. Elles permettent de générer plusieurs demandes d'invoices. Lorsqu'un utilisateur scanne un QR code contenant une *offer*, il envoie un message pour demander une nouvelle invoice au nœud Lightning associé. Le nœud répond avec une invoice que le payeur peut utiliser. Les *offers* permettent ainsi de disposer d'un identifiant unique pour recevoir de nombreux paiements de la part de plusieurs utilisateurs différents sur Lightning.

Termes associés : **BOLT** | **LIGHTNING NETWORK**

OMNI

COUCHE SUPÉRIEURE

Omni Layer (ex-Mastercoin) est une plateforme *open-source* et décentralisée de création et gestion d'actifs sur la blockchain Bitcoin. Parmi les actifs sur Omni on peut citer Tether (USDT) et MaidSafeCoin (MAID).

Termes associés : **COLORED COINS** | **OFF-CHAIN**

ON-CHAIN

PROTOCOLE | FR : SUR-CHAÎNE

Désigne les transactions enregistrées directement sur la blockchain Bitcoin. Ce terme s'oppose à « *off-chain* » qui désigne des opérations qui sont plus ou moins liées avec la blockchain Bitcoin, mais qui se déroulent en dehors de la blockchain principale.

Termes associés : **BLOCKCHAIN** | **OFF-CHAIN**

ON-GRID

MINAGE | FR : RACCORDÉ AU RÉSEAU

Désigne une opération de minage de Bitcoin qui est connectée au réseau électrique général. Les mineurs *on-grid* se branchent sur le réseau électrique classique, utilisé également par les consommateurs résidentiels, pour alimenter leurs ASICs. Cette méthode offre une source d'énergie stable, mais est souvent plus coûteuse comparée au minage *off-grid*, en raison de marges de négociation plus limitées.

Parfois, les fermes de minage *on-grid* participent à des programmes de gestion de la demande. Elles ajustent leur consommation d'énergie en fonction des fluctuations de production et de consommation d'électricité sur le réseau, afin de maintenir l'équilibre et éviter le gaspillage ou le *blackout*.

Termes associés : **BEHIND-THE-METER** | **OFF-GRID**

ONION ADDRESS

RÉSEAU | FR : ADRESSE ONION

Adresse spéciale utilisée pour accéder à un *onion service* sur le réseau Tor. Elle se termine par le suffixe `.onion` et n'est pas résolue par le DNS classique : seul un client connecté au réseau Tor peut l'atteindre. Une adresse onion v3 (le format actuel) est une chaîne de 56 caractères alphanumériques dérivée de la clé publique `Ed25519` du service, ce qui permet au client de vérifier cryptographiquement qu'il communique avec le bon serveur.

Dans l'écosystème Bitcoin, les adresses onion permettent aux nœuds Bitcoin et Lightning d'exposer leurs services sans révéler leur adresse IP réelle. Un nœud configuré avec une adresse `.onion` reste accessible par ses pairs sur le réseau P2P, tout en dissimulant sa localisation géographique et l'identité de son opérateur. Cela renforce la résistance à la censure et la confidentialité, au prix d'une latence légèrement supérieure à celle du *clearnet*.

Termes associés : **CLEARNET** | **ONION SERVICE**

ONION MESSAGE

LIGHTNING NETWORK | FR : MESSAGE EN OIGNON

Messages légers et flexibles transmis entre pairs sur le Lightning Network en utilisant le routage en oignon, sans nécessiter l'envoi de fonds ni le verrouillage de liquidité. Contrairement aux paiements Lightning qui passent par les HTLC, les onion messages sont simplement relayés de pair en pair le long de connexions existantes, ce qui les rend gratuits pour l'expéditeur.

Les onion messages constituent la brique de communication du protocole BOLT-12 (*offers*). Ils permettent à un payeur de contacter un destinataire pour demander une invoice, même sans canal direct entre eux. Grâce aux *blinded paths*, le destinataire peut recevoir ces messages sans révéler son identité ni sa position dans le réseau : il construit une route chiffrée vers lui-même depuis un nœud populaire, ce qui dilue son identité parmi de nombreux candidats possibles.

Au-delà des *offers*, les onion messages ouvrent la voie aux paiements asynchrones : un LSP peut retenir un paiement et utiliser un onion message pour signaler que le destinataire est en ligne, éliminant ainsi l'obligation pour le receveur d'être connecté en permanence.

Termes associés : **BOLT-12** | **ROUTAGE EN OIGNON**

ONION_PRIVATE_KEY

RÉSEAU

Fichier anciennement utilisé dans Bitcoin Core pour stocker une clé privée associée à un service caché Tor V2 pour l'option `-listenonion`. Ce fichier n'est plus utilisé depuis la version 0.21.0 au profit de la V3 de Tor.

Termes associés : **ONION_V3_PRIVATE_KEY** | **TOR**

ONION SERVICE

RÉSEAU | FR : SERVICE CACHÉ

Service hébergé sur le réseau Tor, accessible uniquement via une adresse `.onion`, sans exposer l'adresse IP du serveur. Anciennement appelé « *hidden service* », le terme officiel a été renommé en « *onion service* » par le projet Tor. Le serveur et le client communiquent à travers des circuits Tor, de sorte que ni l'un ni l'autre ne connaît l'adresse IP de son correspondant.

Un onion service fonctionne en publiant des points d'introduction (*introduction points*) dans la table de hachage distribuée du réseau Tor. Le client choisit ensuite un point de rendez-vous (*rendezvous point*) et le communique au service via un point d'introduction. Le client et le serveur construisent chacun un circuit Tor jusqu'à ce point de rendez-vous, afin d'établir une connexion mutuelle sans qu'aucune donnée ne transite en clair sur internet.

Dans le contexte de Bitcoin, de nombreux nœuds Bitcoin Core et Lightning exposent un onion service pour accepter des connexions entrantes de manière anonyme. Cela permet à un opérateur de nœud de participer au réseau P2P sans révéler son adresse IP ni sa localisation, ce qui renforce la résistance à la censure et la confidentialité de l'infrastructure.

Termes associés : **ONION ADDRESS** | **TOR**

ONION_V3_PRIVATE_KEY

RÉSEAU

Fichier utilisé dans Bitcoin Core pour stocker une clé privée associée à un service caché Tor pour l'option `-listenonion`. Lorsque cette option est activée, `bitcoind` crée automatiquement un service caché Tor, permettant au nœud de communiquer sur le réseau Tor.

Termes associés : **ONION_PRIVATE_KEY** | **TOR**

OOR PAYMENT

COUCHE SUPÉRIEURE | FR : PAIEMENT HORS ROUND

Mécanisme de paiement instantané au sein du protocole Ark permettant de transférer des VTXO entre utilisateurs sans attendre le prochain round. Les rounds Ark étant peu fréquents (15 à 60 minutes) et nécessitant de la liquidité de la part de l'ASP, les paiements OOR offrent une alternative rapide et moins coûteuse pour les transferts courants.

Pour effectuer un paiement OOR, l'expéditeur crée une transaction réutilisant la clause de forfait (*forfeit clause*) de la politique du VTXO, ce qui redirige celui-ci vers un nouveau VTXO appartenant au destinataire. L'ASP co-signe cette transaction, puis l'expéditeur la transmet au receveur. Le VTXO OOR obtenu possède les mêmes propriétés qu'un VTXO classique : il peut être utilisé comme entrée dans un round ou servir à un nouveau paiement OOR.

Le modèle de confiance diffère cependant des paiements en round. Le destinataire d'un paiement OOR doit faire confiance à l'expéditeur et à l'ASP pour qu'ils ne s'entendent pas pour effectuer une double dépense. En cas de fraude, les signatures des deux dépenses deviennent publiques, permettant au receveur

de prouver cryptographiquement la violation. L'utilisateur peut à tout moment soumettre son VTXO OOR lors du prochain round pour le rafraîchir en VTXO classique pleinement *trustless*.

Les OOR Payments sont parfois désignés sous le terme « *arkoor* ».

Termes associés : **ARK** **VTXO**

OP_0 - 0X00

SCRIPT

Pousse la valeur 0 sur la pile. Il est souvent utilisé pour représenter la valeur booléenne faux dans les scripts. `OP_0` est également utilisé pour initialiser les scripts.

`OP_0` est identique à `OP_FALSE` (`0X00`) et `OP_PUSHNUM_0`.

Termes associés : **OP_1 - 0X51** **PILE**

OP_0NOTEQUAL - 0X92

SCRIPT

Vérifie si l'élément au sommet de la pile est différent de zéro. Si l'élément est autre que zéro, il pousse `1` (vrai) sur la pile, sinon, il pousse `0` (faux).

Termes associés : **OP_EQUAL - 0X87** **OP_NUMEQUAL - 0X9C**

OP_1 - 0X51

SCRIPT

Pousse la valeur `1` sur la pile. Il est souvent utilisé pour représenter la valeur booléenne vrai dans les scripts.

`OP_1` est identique à `OP_TRUE` (`0X51`) et `OP_PUSHNUM_1`.

Termes associés : **OP_0 - 0X00** **OP_TRUE - 0X51**

OP_1ADD - 0X8B

SCRIPT

Ajoute `1` à la valeur en haut de la pile.

Termes associés : **OP_1SUB - 0X8C** **OP_ADD - 0X93**

OP_1NEGATE - 0X4F

SCRIPT

Pousse la valeur `-1` sur la pile. Cet opcode est utilisé dans les scripts pour représenter la valeur négative `-1`.

Termes associés : **OP_1 - 0X51** **OP_NEGATE - 0X8F**

OP_1SUB - 0X8C

SCRIPT

Soustrait `1` à la valeur en haut de la pile.

Termes associés : **OP_1ADD - 0X8B** **OP_SUB - 0X94**

OP_2 À OP_16 - 0X52 À 0X60

SCRIPT

Les opcodes de `OP_2` jusqu'à `OP_16` poussent les valeurs numériques respectives de 2 à 16 sur la pile. On les utilise pour simplifier les scripts en permettant l'insertion de petites valeurs numériques. Ce type d'opcode est notamment utilisé dans les scripts multisignatures. Voici un exemple de `scriptPubKey` pour un multisig 2/3 :

```
OP_2
<Clé publique A>
<Clé publique B>
<Clé publique C>
OP_3
OP_CHECKMULTISIG
```

Tous ces opcodes sont parfois également nommés `OP_PUSHNUM_N`.

Termes associés : **OP_1 - 0X51** **OP_CHECKMULTISIG - 0XAE**

OP_2DROP - 0XD6

SCRIPT

Supprime les deux éléments en haut de la pile. Autrement dit, `OP_2DROP` supprime le sommet de la pile et le deuxième élément de la pile. Cet opcode est l'équivalent de l'enchainement de deux `OP_DROP`.

Termes associés : **OP_DROP - 0X75** **OP_NIP - 0X77**

OP_2DUP - 0X6E

SCRIPT

Duplique les deux éléments en haut de la pile, puis les place en haut de la pile. Par exemple, si la pile est :

```
A
B
C
D
```

`OP_2DUP` produira :

```
A
B
A
B
C
D
```

Termes associés : **OP_3DUP - 0X6F** **OP_DUP - 0X76**

OP_2OVER - 0X70

SCRIPT

Copie les deux éléments qui se trouvent à la quatrième et à la troisième place en partant du haut de la pile, puis les place en haut de la pile. Par exemple, si la pile est :

```
A
B
C
D
```

`OP_2OVER` produira :

```
C
D
A
B
C
D
```

Termes associés : **OP_2DUP - 0X6E** **OP_OVER - 0X78**

OP_2ROT - 0X71

SCRIPT

Déplace les deux éléments qui se trouvent à la sixième et à la cinquième place du sommet de la pile vers le sommet. Par exemple, si la pile est :

```
A
B
C
D
E
F
```

`OP_2ROT` produira :

```
E
F
A
B
C
D
```

Termes associés : **OP_2SWAP - 0X72** **OP_ROT - 0X7B**

OP_2SWAP - 0X72

SCRIPT

Échange les deux éléments situés au sommet de la pile avec les deux éléments situés juste en dessous d'eux. Par exemple, si la pile est :

```
A
B
C
D
```

`OP_2SWAP` produira :

```
C
D
A
B
```

Termes associés : **OP_2ROT - 0X71** **OP_SWAP - 0X7C**

OP_3DUP - 0X6F

SCRIPT

Duplique les trois éléments en haut de la pile, puis les place en haut de la pile. Par exemple, si la pile est :

```
A
B
C
D
```

`OP_3DUP` produira :

```
A
B
C
A
B
C
D
```

Termes associés : **OP_2DUP - 0X6E** **OP_DUP - 0X76**

OP_ABS - 0X90

SCRIPT

Remplace l'élément supérieur de la pile par sa valeur absolue. Cette opération supprime le signe de l'élément, transformant toute valeur négative en positive, sans changer les valeurs positives.

Termes associés : **OP_NEGATE - 0X8F** **OP_NOT - 0X91**

OP_ADD - 0X93

SCRIPT

Additionne les deux éléments au sommet de la pile. Il prend les deux valeurs en haut de la pile, il les additionne et il les remplace par le résultat.

Termes associés : **OP_1ADD - 0X8B** **OP_SUB - 0X94**

OP_BOOLAND - 0X9A

SCRIPT

Reproduit le comportement d'une porte logique `AND`. Il prend deux valeurs au sommet de la pile et renvoie `1` seulement si les deux valeurs sont non nulles. Dans le cas contraire, il renvoie `0`.

Termes associés : **OP_BOOLOR - 0X9B** **OP_NOT - 0X91**

OP_BOOLOR - 0X9B

SCRIPT

Reproduit le comportement d'une porte logique `OR`. Il prend deux valeurs au sommet de la pile et renvoie `1` si l'un ou l'autre des éléments ou les deux sont non nuls. Dans le cas contraire, il renvoie `0`.

Termes associés : **OP_BOOLAND - 0X9A** **OP_NOT - 0X91**

OP_CAT - 0X7E

SCRIPT

Permet de concaténer les deux éléments en haut de la pile (c'est-à-dire de les mettre bout-à-bout). Cet opcode a été désactivé, il est donc actuellement impossible de l'utiliser. Toutefois, il est récemment revenu sur le devant de la scène. Certains souhaiteraient pouvoir l'ajouter à Tapscript afin de permettre la combinaison d'objets sur la pile, et ainsi améliorer l'expressivité de ce langage. Ce simple outil supplémentaire pourrait permettre :

- L'utilisation des signatures de Lamport qui sont a priori résistantes aux attaques quantiques ;
- La mise en place de Vaults ;
- L'utilisation de covenants ;
- Ou encore, l'utilisation de contrats de non-équivocation.

Termes associés : **COVENANT** **TAPSCRIPT**

OP_CHECKHASHVERIFY - CHV

SCRIPT

Nouvel opcode proposé en 2012 dans le BIP-0017 par Luke Dashjr qui permet de disposer des mêmes fonctionnalités que `OP_EVAL` ou P2SH. Il aurait dû permettre de hacher la fin du `scriptSig`, de comparer le résultat avec le haut de la pile et rendre la transaction invalide si les deux hachages ne correspondaient pas. Cet opcode n'a jamais été implémenté.

Termes associés : **OP_EVAL** **P2SH**

OP_CHECKLOCKTIMEVERIFY - 0XB1

SCRIPT

Rend la transaction invalide sauf si toutes ces conditions sont réunies :

- La pile n'est pas vide ;
- La valeur du haut de la pile est supérieure ou égale à `0` ;
- Le type de timelock est le même entre le champ `nLockTime` et la valeur du

haut de la pile (temps réel ou numéro de bloc) ;
- Le champ `nSequence` de l'input n'est pas égal à `0×ffffffff` ;
- La valeur du champ `nLockTime` de la transaction est supérieure ou égale à la valeur du haut de la pile.

Si une seule de ces conditions n'est pas remplie, le script contenant l'`OP_CHECKLOCKTIMEVERIFY` ne peut être satisfait. Si toutes ces conditions sont valides, alors `OP_CHECKLOCKTIMEVERIFY` agit comme un `OP_NOP`, c'est-à-dire qu'il ne fait aucune action sur le script. C'est un peu comme s'il disparaissait. `OP_CHECKLOCKTIMEVERIFY` impose donc une contrainte de temps sur la dépense de l'UTXO sécurisé avec le script le contenant. Il peut le faire soit sous la forme d'une date exprimée en temps Unix, soit sous la forme d'un numéro de bloc. Pour ce faire, il restreint les valeurs possibles pour le champ `nLockTime` de la transaction qui le dépense, et ce champ `nLockTime` restreint lui-même le moment où la transaction peut être incluse dans un bloc.

Cet opcode est un remplaçant d'`OP_NOP`. Il a été placé sur l'`OP_NOP2`. Il est souvent appelé par son acronyme « CLTV ». Attention, `OP_CLTV` ne doit pas être confondu avec le champ `nLockTime` d'une transaction. Le premier utilise le second, mais leurs natures et leurs actions sont différentes.

Terme associé : **NLOCKTIME**

OP_CHECKMULTISIG - 0XAE

SCRIPT

Vérifie plusieurs signatures contre plusieurs clés publiques. Il prend en entrée une série de `N` clés publiques et `M` signatures, où `M` peut être inférieur ou égal à `N`. `OP_CHECKMULTISIG` vérifie si au moins `M` signatures correspondent à `M` des `N` clés publiques. À noter qu'en raison d'un bug *off-by-one* historique, un élément supplémentaire est supprimé par `OP_CHECKMULTISIG` sur la pile. Cet élément est appelé « *dummy element* ». Pour éviter une erreur dans le `scriptSig`, on inclut donc un `OP_0` qui est un élément inutile afin de satisfaire la suppression et outrepasser le bug. Depuis le BIP-0147 (introduit avec SegWit en 2017), l'élément inutile consommé à cause du bug doit forcément être `OP_0` pour que le script soit valide, car c'était un vecteur de malléabilité. Cet opcode a été supprimé dans Tapscript.

Termes associés : **OP_CHECKMULTISIGVERIFY - 0XAF** **TAPSCRIPT**

OP_CHECKMULTISIGVERIFY - 0XAF

SCRIPT

Combine un `OP_CHECKMULTISIG` et un `OP_VERIFY`. Il prend plusieurs signatures et clés publiques pour vérifier que `M` parmi `N` signatures sont valides, comme le fait `OP_CHECKMULTISIG`. Puis, à l'instar d'`OP_VERIFY`, si la vérification échoue, le script s'arrête immédiatement avec une erreur. Si la vérification est réussie, le script continue sans pousser de valeur sur la pile. Cet opcode a été supprimé dans Tapscript.

Termes associés : **OP_CHECKMULTISIG - 0XAE** **OP_VERIFY - 0X69**

OP_CHECKSEQUENCEVERIFY - 0XB2

SCRIPT

Rend la transaction invalide si une seule de ces caractéristiques est observée :

- La pile est vide ;
- La valeur du haut de la pile est inférieure à `0` ;
- L'indicateur de désactivation de la valeur en haut de la pile est non défini et ; La version de la transaction est inférieure à `2` ou ; L'indicateur de désactivation du champ `nSequence` de l'input est défini ou ; Le type de timelock n'est pas le même entre le champ `nSequence` et la valeur du haut de la pile (temps réel ou nombre de blocs) ou ; La valeur en haut de la pile est supérieure à la valeur du champ `nSequence` de l'input.

Si une ou plusieurs de ces caractéristiques est observée, le script contenant l'`OP_CHECKSEQUENCEVERIFY` ne peut être satisfait. Si toutes les conditions sont valides, alors `OP_CHECKSEQUENCEVERIFY` agit comme un `OP_NOP`, c'est-à-dire qu'il ne fait aucune action sur le script. C'est un peu comme s'il disparaissait. `OP_CHECKSEQUENCEVERIFY` impose donc une contrainte de temps relative sur la dépense de l'UTXO sécurisé avec le script le contenant. Il peut le faire soit sous la forme d'un temps réel, soit sous la forme d'un nombre de blocs. Pour ce faire, il restreint les valeurs possibles pour le champ `nSequence` de l'input qui le dépense, et ce champ `nSequence` restreint lui-même le moment où la transaction qui comprend cet input peut être incluse dans un bloc.

Cet opcode est un remplaçant d'`OP_NOP`. Il a été placé sur l'`OP_NOP3`. Il est souvent appelé par son acronyme « CSV ». Attention, `OP_CSV` ne doit pas être confondu avec le champ `nSequence` d'une transaction. Le premier utilise le second, mais leurs natures et leurs actions sont différentes.

Terme associé : **NSEQUENCE**

OP_CHECKSIG - 0XAC

SCRIPT

Vérifie la validité d'une signature par rapport à une clé publique donnée. Il prend les deux éléments du sommet de la pile : la signature et la clé publique, et évalue si la signature est correcte pour le hachage de la transaction et la clé publique spécifiée. Si la vérification est réussie, il pousse la valeur `1` (vrai) sur la pile, sinon `0` (faux). Cet opcode a été modifié dans Tapscript afin de vérifier des signatures de Schnorr.

Termes associés : **OP_CHECKSIGVERIFY - 0XAD** **SCHNORR**

OP_CHECKSIGADD - 0XBA

SCRIPT

Extrait les trois valeurs en haut de la pile : une `clé publique`, un `CScriptNum n` et une `signature`. Si la signature n'est pas le vecteur vide et n'est pas valide, le script se termine avec une erreur. Si la signature est valide ou est le vecteur vide (`OP_0`), deux cas de figure se présentent :

- Si la signature est le vecteur vide : un `CScriptNum` avec la valeur de `n` est

poussé sur la pile et l'exécution continue ;
- Si la signature n'est pas le vecteur vide et demeure valide : un `CScriptNum` avec la valeur de `n + 1` est poussé sur la pile et l'exécution continue.

Pour vulgariser, `OP_CHECKSIGADD` effectue une opération similaire à `OP_CHECKSIG`, mais au lieu de pousser vrai ou faux sur la pile, il ajoute `1` à la deuxième valeur en haut de la pile si la signature est valide, ou laisse cette valeur inchangée si la signature représente le vecteur vide. `OP_CHECKSIGADD` permet de créer les mêmes politiques multisignatures dans Tapscript qu'avec `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY` mais de manière vérifiable par lots, c'est-à-dire qu'il supprime le processus de recherche dans la vérification d'un multisig traditionnel et accélère donc la vérification tout en réduisant la charge opérationnelle sur les CPU des nœuds. Cet opcode a été ajouté dans Tapscript uniquement pour les besoins de Taproot.

Termes associés : **OP_CHECKMULTISIG - 0XAE** | **OP_CHECKSIG - 0XAC**

OP_CHECKSIGFROMSTACK - 0XCC

SCRIPT

Opcode `0×cc` proposé dans le BIP-0348 pour Tapscript, en remplacement de `OP_SUCCESS204`. Contrairement à `OP_CHECKSIG`, qui vérifie une signature sur un hash dérivé de la transaction courante, `OP_CHECKSIGFROMSTACK` permet de vérifier une signature Schnorr sur un message arbitraire fourni directement sur la pile.

L'exécution commence par la lecture de trois éléments depuis le sommet de la pile : la clé publique (sommet), le message (deuxième position) et la signature (troisième position). Ces trois éléments sont ensuite retirés de la pile. Si la clé publique est vide, le script échoue immédiatement. Si elle fait 32 octets, elle est interprétée comme une clé BIP-0340 : une signature non vide est alors validée selon le schéma Schnorr contre cette clé et le message, et toute signature invalide provoque l'arrêt du script. Une signature vide pousse un vecteur vide sur la pile et l'exécution continue normalement. Une signature valide pousse la valeur `0×01`. Les clés d'une longueur différente de 0 et de 32 octets sont traitées comme des types inconnus, et leur vérification réussit systématiquement afin de permettre de futures extensions.

Chaque vérification avec une signature non vide est comptabilisée dans le budget de sigops défini par le BIP-0342. Cette capacité à authentifier des données arbitraires ouvrirait des possibilités comme l'introspection de transaction, la délégation de droits de dépense et la construction de canaux Lightning Symmetry. Cette proposition est pour le moment toujours en brouillon, donc non implémentée sur Bitcoin.

Termes associés : **BIP-0348** | **OP_CHECKSIG - 0XAC**

OP_CHECKSIGVERIFY - 0XAD

SCRIPT

Effectue la même opération que `OP_CHECKSIG`, mais si la vérification de la signature échoue, le script s'arrête immédiatement avec une erreur et la transaction est donc invalide. Si la vérification réussit, le script continue sans pousser de valeur `1` (vrai) sur la pile. Pour résumer, `OP_CHECKSIGVERIFY` réalise l'opération `OP_CHECKSIG` suivie de `OP_VERIFY`. Cet opcode a été modifié dans Tapscript afin de vérifier des signatures de Schnorr.

Termes associés : **OP_CHECKSIG - 0XAC** **OP_VERIFY - 0X69**

OP_CHECKTEMPLATEVERIFY - 0XB3

SCRIPT

Opcode `0×b3` proposé dans le BIP-0119, en remplacement de `OP_NOP4`. Il impose que la transaction de dépense corresponde exactement à un gabarit prédéfini, ce qui en fait un mécanisme de covenant non récursif.

L'opcode lit l'élément au sommet de la pile sans le retirer. Si cet élément fait 32 octets, il est comparé au `DefaultCheckTemplateVerifyHash` calculé à partir de la transaction courante et de l'index de l'input en cours d'exécution. Ce hash est le `SHA256` de la concaténation sérialisée de : la version de la transaction, le locktime, le hash des scriptSigs (s'ils sont non vides), le nombre d'inputs, le hash des séquences, le nombre d'outputs, le hash des outputs et l'index de l'input courant. Si les deux valeurs ne correspondent pas, le script échoue. Si l'élément ne fait pas 32 octets, l'opcode se comporte comme un `OP_NOP` afin de permettre de futures extensions.

En contraignant la quasi-totalité des champs de la transaction de dépense, CTV permet de spécifier à l'avance les sorties exactes sans recourir à des signatures ou à des clés éphémères. Cette propriété ouvre la voie à des applications telles que les arbres de paiement pour le contrôle de congestion, les *vaults*, les joinpools ou l'amélioration de l'efficacité des DLC. Cette proposition est pour le moment toujours en brouillon.

Termes associés : **BIP-0119** **COVENANT**

OP_CODESEPARATOR - 0XAB

SCRIPT

Modifie le script de sortie en cours, en indiquant que seules les opérations qui suivent cet opcode seront prises en compte dans la vérification des signatures des entrées correspondantes. Cela permet de segmenter un script complexe en plusieurs parties, où chaque segment peut être signé indépendamment.

Termes associés : **OP_CHECKSIG - 0XAC** **SCRIPTPUBKEY**

OP_DEPTH - 0X74

SCRIPT

Pousse le nombre d'éléments dans la pile sur la pile elle-même. Si la pile contient `N` éléments, `OP_DEPTH` ajoutera le nombre `N` en tant que nouvel élément en haut de la pile.

Termes associés : **OP_SIZE - 0X82** **PILE**

OP_DROP - 0X75

SCRIPT

Supprime l'élément situé au sommet de la pile. `OP_DROP` permet d'enlever des données devenues inutiles au cours de l'exécution du script.

Termes associés : **OP_2DROP - 0XD6** **PILE**

OP_DUP - 0X76

SCRIPT

Duplique le sommet de la pile. L'élément en haut de la pile est donc copié et la copie est placée en haut de la pile.

Termes associés : **OP_2DUP - 0X6E** **P2PKH**

OP_ELSE - 0X67

SCRIPT

Modifie le flux d'exécution dans un script conditionnel : il indique que les opérations qui le suivent doivent être exécutées si la condition précédente spécifiée par un `OP_IF` ou un `OP_NOTIF` n'est pas remplie.

Termes associés : **OP_ENDIF - 0X68** **OP_IF - 0X63**

OP_ENDIF - 0X68

SCRIPT

Marque la fin d'une structure de contrôle conditionnelle initiée par un `OP_IF` ou un `OP_NOTIF`, normalement suivis par un ou plusieurs `OP_ELSE`. Il indique que l'exécution du script doit continuer au-delà de la structure conditionnelle, quelle que soit la branche qui a été exécutée. Autrement dit, `OP_ENDIF` permet de délimiter la fin des blocs conditionnels dans les scripts.

Termes associés : **OP_ELSE - 0X67** **OP_IF - 0X63**

OP_EQUAL - 0X87

SCRIPT

Compare les deux valeurs les plus hautes de la pile et pousse `1` sur la pile si elles sont égales, sinon pousse `0`. `OP_EQUAL` permet de vérifier l'égalité de données dans un script.

Termes associés : **OP_EQUALVERIFY - 0X88** **OP_NUMEQUAL - 0X9C**

OP_EQUALVERIFY - 0X88

SCRIPT

Combine les fonctions de `OP_EQUAL` et `OP_VERIFY`. Il vérifie d'abord l'égalité des deux valeurs supérieures de la pile, puis exige que le résultat soit non nul, faute de quoi la transaction est invalide. `OP_EQUALVERIFY` est notamment utilisé dans les scripts de vérification d'adresse.

Termes associés : **OP_EQUAL - 0X87** **OP_VERIFY - 0X69**

OP_EVAL

SCRIPT

Opcode proposé par Gavin Andresen en 2011. Il prend le script situé au sommet de la pile, l'exécute comme s'il faisait partie du `scriptPubKey`, et place son résultat sur la pile. `OP_EVAL` a été abandonné en raison de préoccupations liées à la complexité de cet opcode, qui sera finalement supplanté par P2SH.

Termes associés : **P2SH** **SCRIPTPUBKEY**

OP_FALSE - 0X00

SCRIPT

Identique à `OP_0`.

Termes associés : **OP_0 - 0X00** **OP_TRUE - 0X51**

OP_FROMALTSTACK - 0X6C

SCRIPT

Retire le sommet de la pile alternative (*alt stack*) et le place sur le sommet de la pile principale (*main stack*). Cet opcode est utilisé pour récupérer des données stockées temporairement sur la pile alternative. Pour simplifier, c'est l'opération inverse de `OP_TOALTSTACK`.

Termes associés : **OP_TOALTSTACK - 0X6B** **PILE**

OP_GREATERTHAN - 0XA0

SCRIPT

Compare les deux éléments au sommet de la pile et vérifie si le premier élément est supérieur au deuxième. Si le premier élément est supérieur au deuxième, il pousse `1` (vrai) sur la pile, sinon, il pousse `0` (faux).

Termes associés : **OP_GREATERTHANOREQUAL - 0XA2** **OP_LESSTHAN - 0X9F**

OP_GREATERTHANOREQUAL - 0XA2

SCRIPT

Compare les deux éléments au sommet de la pile et vérifie si le second élément est supérieur ou égal au premier. Si le second élément est supérieur ou égal au premier, il pousse `1` (vrai) sur la pile, sinon, il pousse `0` (faux).

Termes associés : **OP_GREATERTHAN - 0XA0** **OP_LESSTHANOREQUAL - 0XA1**

OP_HASH160 - 0XA9

SCRIPT

Prend l'élément en haut de la pile et le remplace par son hachage en utilisant successivement les fonctions `SHA256` puis `RIPEMD160`. Ce processus en deux étapes génère une empreinte de 160 bits.

Termes associés : **RIPEMD160** **SHA256**

OP_HASH256 - 0XAA

SCRIPT

Prend l'élément en haut de la pile et le remplace par son hachage en utilisant deux fois la fonction `SHA256`. L'entrée est hachée une première fois avec `SHA256` et le condensat est haché une seconde fois avec `SHA256`. Ce processus en deux étapes génère une empreinte de 256 bits.

Terme associé : **SHA256**

OP_IF - 0X63

SCRIPT

Exécute la portion suivante du script si la valeur au sommet de la pile est non nulle (vraie). Si la valeur est nulle (fausse), ces opérations sont sautées, passant directement à celles après `OP_ELSE`, s'il est présent. `OP_IF` permet de lancer une structure de contrôle conditionnelle dans un script. Il détermine le flux de contrôle dans un script en fonction d'une condition fournie au moment de l'exécution de la transaction. `OP_IF` s'utilise avec `OP_ELSE` et `OP_ENDIF` de la manière suivante :

```
<condition>
OP_IF
<opérations si la condition est vraie>
OP_ELSE
<opérations si la condition est fausse>
OP_ENDIF
```

Termes associés : **OP_ELSE - 0X67** **OP_ENDIF - 0X68**

OP_IFDUP - 0X73

SCRIPT

Duplique le sommet de la pile si celui-ci est non nul. Si la valeur en haut de la pile est vraie (c'est-à-dire non nulle), cette valeur est dupliquée sur la pile ; sinon, la pile reste inchangée.

Termes associés : **OP_DUP - 0X76** **OP_IF - 0X63**

OP_INTERNALKEY - 0XCB

SCRIPT

Opcode proposé dans le BIP-0349 pour être ajouté à Tapscript (`0×cb`), qui pousse la clé interne Taproot (*internal key*) sur la pile. Concrètement, lorsqu'un script dans une feuille Taproot a besoin de connaître la clé publique qui a servi

à construire la sortie, il peut appeler `OP_INTERNALKEY` au lieu de devoir inclure cette clé en dur dans le script. Le résultat est la représentation *x-only* de 32 octets de la clé interne.

Termes associés : **BIP-0349** **TAPROOT**

OP_LESSTHAN - 0X9F

SCRIPT

Compare les deux éléments au sommet de la pile et vérifie si le second élément est inférieur au premier. Si le second élément est inférieur au premier, il pousse `1` (vrai) sur la pile, sinon, il pousse `0` (faux).

Termes associés : **OP_GREATERTHAN - 0XA0** **OP_LESSTHANOREQUAL - 0XA1**

OP_LESSTHANOREQUAL - 0XA1

SCRIPT

Compare les deux éléments au sommet de la pile et vérifie si le second élément est inférieur ou égal au premier. Si le second élément est inférieur ou égal au premier, il pousse `1` (vrai) sur la pile, sinon, il pousse `0` (faux).

Termes associés : **OP_GREATERTHANOREQUAL - 0XA2** **OP_LESSTHAN - 0X9F**

OP_MAX - 0XA4

SCRIPT

Sélectionne le plus grand des deux éléments en haut de la pile et le pousse sur la pile. Cette opération conserve uniquement la plus grande des deux valeurs au sommet.

Termes associés : **OP_GREATERTHAN - 0XA0** **OP_MIN - 0XA3**

OP_MIN - 0XA3

SCRIPT

Sélectionne le plus petit des deux éléments en haut de la pile et le pousse sur la pile. Cette opération conserve uniquement la plus petite des deux valeurs au sommet.

Termes associés : **OP_LESSTHAN - 0X9F** **OP_MAX - 0XA4**

OP_NEGATE - 0X8F

SCRIPT

Inverse le signe de l'élément supérieur de la pile. Si la valeur est positive, elle devient négative, et inversement.

Termes associés : **OP_ABS - 0X90** **OP_SUB - 0X94**

OP_NIP - 0X77

SCRIPT

Supprime l'élément situé juste en dessous du sommet de la pile (le second en partant du haut).

Termes associés : **OP_DROP - 0X75** **OP_OVER - 0X78**

OP_NOP - 0X61

SCRIPT

Ne produit aucun effet sur la pile ou l'état du script. Il ne fait aucun déplacement, aucune vérification, ni aucune modification. Il ne fait juste rien, à moins que l'on ait décidé l'inverse via un soft fork. En effet, depuis leurs modifications par Satoshi Nakamoto en 2010, les commandes `OP_NOP` (`OP_NOP1` (`0XB0`) jusqu'à `OP_NOP10` (`0XB9`)) sont utilisées pour ajouter de nouveaux opcodes sous forme de soft fork.

Ainsi, l'`OP_NOP2` (`0XB1`) et l'`OP_NOP3` (`0XB2`) ont déjà été utilisés pour implémenter respectivement l'`OP_CHECKLOCKTIMEVERIFY` et l'`OP_CHECKSEQUENCEVERIFY`. On les utilise en combinaison avec `OP_DROP` afin de supprimer de la pile les valeurs temporelles associées, et ainsi pouvoir continuer l'exécution du script, que le nœud soit à jour ou non. Les `OP_NOP` permettent donc d'insérer un point d'interruption dans un script pour vérifier des conditions supplémentaires déjà existantes ou pouvant être ajoutées avec de futurs soft fork. Depuis Tapscript, l'utilisation des `OP_NOP` a été remplacée par l'utilisation des `OP_SUCCESS` étant plus efficace.

Termes associés : **OP_CHECKLOCKTIMEVERIFY - 0XB1** **OP_CHECKSEQUENCEVERIFY - 0XB2**

OP_NOT - 0X91

SCRIPT

Inverse la valeur booléenne du sommet de la pile : si cette valeur est non nulle, l'opérateur la remplace par 0, sinon par 1.

Termes associés : **OP_0NOTEQUAL - 0X92** **OP_NEGATE - 0X8F**

OP_NOTIF - 0X64

SCRIPT

Fonctionne de manière opposée à `OP_IF`, exécutant la portion suivante du script si la valeur au sommet de la pile est nulle (fausse).

Termes associés : **OP_ELSE - 0X67** **OP_IF - 0X63**

OP_NUMEQUAL - 0X9C

SCRIPT

Compare les deux éléments au sommet de la pile pour vérifier s'ils sont numériquement égaux. Si les valeurs sont égales, il pousse `1` (vrai) sur la pile, sinon, il pousse `0` (faux).

Termes associés : **OP_NUMEQUALVERIFY - 0X9D** **OP_NUMNOTEQUAL - 0X9E**

OP_NUMEQUALVERIFY - 0X9D

SCRIPT

Combine les opérations `OP_NUMEQUAL` et `OP_VERIFY`. Il compare numériquement les deux éléments au sommet de la pile. Si les valeurs sont égales, `OP_NUMEQUALVERIFY` supprime le résultat vrai de la pile et continue l'exécution du script. Si les valeurs ne sont pas égales, le résultat est faux et le script échoue immédiatement.

Termes associés : **OP_NUMEQUAL - 0X9C** **OP_VERIFY - 0X69**

OP_NUMNOTEQUAL - 0X9E

SCRIPT

Compare les deux éléments au sommet de la pile pour vérifier s'ils sont numériquement non égaux. Si les valeurs ne sont pas égales, il pousse `1` (vrai) sur la pile, sinon, il pousse `0` (faux). C'est l'inverse de `OP_NUMEQUAL`.

Termes associés : **OP_NUMEQUAL - 0X9C** **OP_NUMEQUALVERIFY - 0X9D**

OP_OVER - 0X78

SCRIPT

Duplique le deuxième élément à partir du haut de la pile et le place sur le haut de la pile.

Termes associés : **OP_2OVER - 0X70** **PILE**

OP_PICK - 0X79

SCRIPT

Duplique un élément de la pile et le place en haut de la pile, en fonction de la profondeur spécifiée par la valeur en haut de la pile avant l'opération. Par exemple, si la valeur en haut de la pile est `4`, `OP_PICK` va dupliquer le cinquième élément de la pile en partant du sommet (l'indexation commence à `0`), et il va placer cette copie au sommet.

Termes associés : **OP_ROLL - 0X7A** **PILE**

OP_PUSHDATA1 - 0X4C

SCRIPT

Pousse une certaine quantité de données sur la pile. Il est suivi d'un octet qui indique la longueur des données à pousser (jusqu'à 255 octets). Cet opcode est utilisé pour inclure des données de taille variable dans les scripts.

Termes associés : **OCTET** **OP_PUSHDATA2 - 0X4D**

OP_PUSHDATA2 - 0X4D

SCRIPT

Permet de pousser une grande quantité de données sur la pile. Il est suivi de deux octets (petit-boutistes) qui spécifient la longueur des données à pousser (jusqu'à 65 535 octets). Il est utilisé pour insérer des données plus volumineuses dans les scripts.

Termes associés : **OP_PUSHDATA1 - 0X4C** **OP_PUSHDATA4 - 0X4E**

OP_PUSHDATA4 - 0X4E

SCRIPT

Permet de pousser une très grande quantité de données sur la pile. Il est suivi de quatre octets (petit-boutistes) qui indiquent la longueur des données à pousser (jusqu'à environ 4,3 Go). Cet opcode est utilisé pour insérer de très grandes données dans les scripts, bien que son usage soit extrêmement rare en raison des limitations pratiques de la taille des transactions.

Termes associés : **OP_PUSHDATA1 - 0X4C** **OP_PUSHDATA2 - 0X4D**

OP_RETURN - 0X6A

SCRIPT

Signale un script invalide, ce qui rend l'output qui le contient non dépensable de manière prouvée. Les nœuds du réseau peuvent donc supprimer cet output de leurs UTXO set. `OP_RETURN` est souvent utilisé pour inscrire des données arbitraires dans la blockchain.

Termes associés : **NULL DATA** **UTXO**

OP_RIPEMD160 - 0XA6

SCRIPT

Prend l'élément en haut de la pile et le remplace par son hachage en utilisant la fonction `RIPEMD160`.

Terme associé : **RIPEMD160**

OP_ROLL - 0X7A

SCRIPT

Déplace un élément de la pile en haut de la pile, en fonction de la profondeur spécifiée par la valeur en haut de la pile avant l'opération. Par exemple, si la valeur en haut de la pile est `4`, `OP_ROLL` va sélectionner le cinquième élément de la pile en partant du sommet (l'indexation commence à `0`), et il va déplacer cette valeur au sommet. `OP_ROLL` joue le même rôle que `OP_PICK`, mis à part qu'il retire l'élément de sa position initiale.

Termes associés : **OP_PICK - 0X79** **PILE**

OP_ROT - 0X7B

SCRIPT

Déplace au sommet de la pile le troisième élément à partir du sommet de la pile. Les deux éléments qui étaient au-dessus de lui sont poussés en dessous en conservant leur ordre relatif.

Termes associés : OP_2ROT - 0X71 | OP_SWAP - 0X7C

OP_SHA1 - 0XA7

SCRIPT

Prend l'élément en haut de la pile et le remplace par son hachage en utilisant la fonction `SHA1`. L'utilisation de cette fonction est aujourd'hui déconseillée dans un cadre sécurisé.

Termes associés : FONCTION DE HACHAGE | OP_SHA256 - 0XA8

OP_SHA256 - 0XA8

SCRIPT

Prend l'élément en haut de la pile et le remplace par son hachage en utilisant la fonction `SHA256`.

Terme associé : SHA256

OP_SIZE - 0X82

SCRIPT

Mesure la taille en nombre d'octets de l'élément en haut de la pile et renvoie cette taille au sommet de la pile, sans pour autant modifier l'élément analysé en lui-même.

Termes associés : OCTET | PILE

OP_SUB - 0X94

SCRIPT

Soustrait les deux éléments au sommet de la pile. Il prend les deux valeurs en haut de la pile, il les soustrait et il les remplace par le résultat.

Termes associés : OP_ADD - 0X93 | PILE

OP_SUCCESS

SCRIPT

Les `OP_SUCCESS` représentent une série d'opcodes qui ont été désactivés par le passé et qui sont dorénavant réservés pour une utilisation future dans Tapscript. Leur but final est de faciliter des mises à jour et des extensions du langage script, en permettant l'introduction de nouvelles fonctionnalités via des soft forks. Lorsqu'un de ces opcodes est rencontré dans un script, il entraîne la réussite immédiate et inconditionnelle de l'ensemble du script, peu importe les données ou les conditions présentes. L'exécution ne se poursuit pas : le script est directement considéré comme valide.

Ainsi, lorsque l'on ajoute un nouvel opcode sur un `OP_SUCCESS`, cela représente forcément l'ajout d'une règle plus restrictive que la règle précédente. Les

nœuds à jour peuvent alors vérifier le respect de cette règle et les nœuds pas à jour ne refuseront pas les transactions associées et les blocs qui les incluent, car l'`OP_SUCCESS` valide cette partie du script. Il n'y a donc pas de hard fork.

En comparaison, les `OP_NOP` (*No Operation*) servent également de marqueurs de place dans le script, mais ils n'ont aucun effet sur l'exécution du script. Lorsqu'un `OP_NOP` est rencontré, le script continue simplement sans modifier l'état de la pile ou le déroulement du script. La différence clé est donc dans leur impact sur l'exécution : `OP_SUCCESS` entraîne la réussite immédiate de l'ensemble du script, tandis que `OP_NOP` est neutre, et n'affecte ni la pile ni le flux du script. Ces opcodes sont désignés par `OP_SUCCESSN` où `N` est un numéro permettant de différencier les `OP_SUCCESS`.

Termes associés : **OP_NOP - 0X61** **TAPSCRIPT**

OP_SWAP - 0X7C

SCRIPT

Échange les deux éléments en haut de la pile. L'élément qui était au sommet est déplacé en deuxième position, et l'élément qui était en deuxième position est placé au sommet de la pile.

Termes associés : **OP_2SWAP - 0X72** **PILE**

OP_TOALTSTACK - 0X6B

SCRIPT

Prend le sommet de la pile principale (*main stack*) et le déplace vers la pile alternative (*alt stack*). Cet opcode permet de stocker temporairement des données à part pour une utilisation ultérieure dans le script. L'élément déplacé est donc supprimé de la pile principale. On utilisera ensuite `OP_FROMALTSTACK` pour le remettre au sommet de la pile principale.

Termes associés : **OP_FROMALTSTACK - 0X6C** **PILE**

OP_TRUE - 0X51

SCRIPT

Identique à `OP_1`. Il pousse la valeur `1` sur la pile. Il est souvent utilisé pour représenter la valeur booléenne vrai dans les scripts.

Termes associés : **OP_1 - 0X51** **OP_FALSE - 0X00**

OP_TUCK - 0X7D

SCRIPT

Copie l'élément situé au sommet de la pile et l'insère entre le deuxième élément et le troisième élément de la pile. Par exemple, si la pile est :

```
A
B
C
D
```

`OP_TUCK` va dupliquer le sommet `A` et le placer en troisième position. La pile

en sortie sera :

```
A
B
A
C
D
```

Termes associés : OP_SWAP - 0X7C | PILE

OP_TXHASH - 0XBD

SCRIPT

Opcode `0×bd` proposé dans le BIP-0346 pour Tapscript, en remplacement de `OP_SUCCESS189`. Il fournit un mécanisme généraliste d'introspection de transaction qui produit un hash configurable à partir de certains champs de la transaction de dépense.

L'opcode retire du sommet de la pile un élément appelé `TxFieldSelector`, une structure sérialisée qui détermine quels champs de la transaction doivent être inclus dans le hash. Ce sélecteur permet de cibler, au choix : des champs globaux (version, locktime, index de l'input courant), des champs par input (prevouts, séquences, scriptSigs, valeurs et scriptPubkeys des UTXO dépensés, annexes Taproot) et des champs par output (scriptPubkeys, valeurs). Il offre aussi plusieurs modes de sélection des inputs et outputs : tous, uniquement celui de l'index courant, les N premiers, ou un ensemble d'indices individuels. Après validation du sélecteur, l'opcode pousse sur la pile le hash `SHA256` de 32 octets calculé à partir des champs sélectionnés.

Avec un `TxFieldSelector` vide, `OP_TXHASH` produit un hash sémantiquement équivalent à celui de `OP_CHECKTEMPLATEVERIFY`, ce qui en fait une généralisation stricte du BIP-0119. Combiné avec `OP_CHECKSIGFROMSTACK`, il permettrait de construire des hashs de signature arbitraires et d'émuler `SIGHASH_ANYPREVOUT` ainsi que d'autres modes de signature. Cette proposition est pour le moment toujours en brouillon.

Termes associés : BIP-0346 | OP_CHECKTEMPLATEVERIFY - 0XB3

OP_VAULT - 0XBB

SCRIPT

Opcode `0×bb` proposé dans le BIP-0345 pour Tapscript, en remplacement de `OP_SUCCESS187`. Il constitue le mécanisme d'un système de vault à covenant spécialisé, qui permet d'imposer un délai configurable avant toute dépense, avec possibilité de récupération en cas de compromission.

L'opcode consomme plusieurs éléments depuis la pile (du sommet vers le bas) : un `<leaf-update-script-body>` qui contient le corps du script de remplacement, un `<push-count>` qui indique le nombre d'éléments de données supplémentaires, les éléments de données correspondants, un `<trigger-vout-idx>` qui désigne l'index de la sortie de déclenchement, un `<revault-vout-idx>` qui désigne l'index d'une sortie optionnelle de « revault » (`-1` en son absence), et

un `<revault-amount>` qui spécifie le montant en satoshis à replacer dans le vault. À partir de ces éléments, l'opcode construit un script de remplacement en préfixant le `<leaf-update-script-body>` avec les données fournies, puis vérifie que la sortie de déclenchement possède un taptree identique à celui de l'input courant, à l'exception de la feuille en cours d'exécution qui est remplacée par ce nouveau script. Cette substitution de feuille Taproot préserve toutes les autres feuilles, y compris le chemin de récupération via `OP_VAULT_RECOVER` (`0×bc`).

L'opcode enregistre des vérifications différées pour s'assurer que la valeur de l'input est intégralement reportée entre la sortie de déclenchement et la sortie de « revault » éventuelle. Ce mécanisme permet de regrouper plusieurs inputs de vault dans une même transaction. En cas de succès, la valeur `0×01` est poussée sur la pile. Le BIP-0345 a été fermé et remplacé par le BIP-0443 (`OP_CHECKCONTRACTVERIFY`), qui est plus généraliste.

Termes associés : **BIP-0345** **VAULT**

OP_VER - 0X62

SCRIPT

Permettait de pousser la version du protocole sur la pile. Cet opcode a été désactivé, car s'il avait été utilisé, chaque mise à jour aurait conduit à un hard fork. Le BIP-0342 a modifié cet opcode en `OP_SUCCESS`.

Termes associés : **OP_NOP - 0X61** **OP_SUCCESS**

OP_VERIFY - 0X69

SCRIPT

Exige que la valeur du sommet de la pile soit non nulle (vraie). La transaction est invalide si ce n'est pas le cas. `OP_VERIFY` est utilisé pour confirmer les conditions des scripts.

Termes associés : **OP_CHECKSIGVERIFY - 0XAD** **OP_EQUALVERIFY - 0X88**

OP_WITHIN - 0XA5

SCRIPT

Vérifie si le premier élément en haut de la pile se trouve dans l'intervalle défini par les deuxième et troisième éléments supérieurs. Autrement dit, `OP_WITHIN` vérifie si le premier élément est supérieur ou égal au deuxième et inférieur au troisième. Si cette condition est vraie, il pousse `1` (vrai) sur la pile, sinon, il pousse `0` (faux).

Termes associés : **OP_GREATERTHANOREQUAL - 0XA2** **OP_LESSTHAN - 0X9F**

OPCODES

SCRIPT | FR : CODES OPÉRATOIRES

Ensemble des commandes utilisées dans le système script de Bitcoin. Script est un langage de programmation à piles utilisé pour établir des conditions de dépense, et donc, indirectement, sécuriser des bitcoins. Les instructions utilisées en langage script sont appelées « opcodes ». Ce sont des opérateurs logiques et des commandes pour manipuler les piles (*stacks*). Ces instructions spécifiques sont exécutées par les nœuds du réseau lors de la validation d'une transaction. Script est un langage non-Turing complet. Il peut être catégorisé comme un langage de niveau intermédiaire (presque bas niveau) inspiré du Forth.

Termes associés : **PILE** **SCRIPT**

OPENDIME

PORTEFEUILLE

Dispositif USB à usage unique fabriqué par Coinkite, conçu comme un actif au porteur pour des bitcoins. L'Opendime génère une clé privée en interne lors de sa première utilisation. Cette clé n'est connue de personne. On peut ensuite envoyer des bitcoins sur l'adresse associée. Pour dépenser les bitcoins sécurisés par cette clé, il faut desceller physiquement le dispositif en insérant une pointe dans un trou prévu à cet effet sur le circuit imprimé. La clé privée est alors révélée au format WIF et l'utilisateur peut dépenser les fonds associés. Tant que le dispositif reste scellé, il peut être transmis de main en main sans générer de transaction on-chain. Le scellement physique permet de vérifier avec certitude, lors d'un échange, que la clé privée n'est pas connue du payeur, simplement en inspectant le dispositif.

Termes associés : **CASASCIUS** **COINKITE** **SATSCARD**

OPENNODE

ORGANISATION

Plateforme de traitement de paiements en bitcoins. OpenNode fournit une API qui permet aux commerçants et aux développeurs d'accepter des paiements en bitcoin, à la fois sur la chaîne principale et via le Lightning Network. La plateforme propose des pages de paiement, des liens de facturation et des outils d'intégration. Les commerçants utilisant ce service ont le choix de récupérer leurs fonds en bitcoins ou directement en monnaie fiat sur leur compte bancaire. À noter qu'OpenNode n'est pas un terminal de paiement en *self-custody*, contrairement à BTCPayServer par exemple.

Terme associé : **LIGHTNING NETWORK**

OPENSATS

ORGANISATION

Organisation à but non lucratif fondée en 2021, dédiée au financement de projets libres et *open source* liés à Bitcoin. Structurée en tant qu'association caritative aux États-Unis, OpenSats distribue des subventions à des développeurs et à des projets qui contribuent à l'écosystème Bitcoin et à ses technologies adjacentes.

Terme associé : **BITCOIN CORE**

OPENTIMESTAMPS

OUTIL

Protocole *open source* d'horodatage décentralisé, développé par Peter Todd. OpenTimestamps utilise la blockchain Bitcoin pour prouver l'existence d'un document ou d'une donnée à un instant précis. Le protocole calcule une empreinte cryptographique (*hash*) du fichier à horodater, puis ancre cette empreinte dans une transaction Bitcoin par l'intermédiaire d'un arbre de Merkle. Le résultat est une preuve vérifiable et inaltérable attestant que la donnée existait au moment où le bloc Bitcoin correspondant a été miné. Ce système est utile pour la propriété intellectuelle, les contrats, les preuves légales ou toute situation qui nécessite une preuve temporelle.

Termes associés : **BLOCKCHAIN** **TAPROOT**

ORACLE

COUCHE SUPÉRIEURE

Source d'informations tierce qui fournit des données du monde réel pouvant être interprétées sur Bitcoin. Les oracles permettent aux contrats intelligents, tels que les DLC, d'exécuter des conditions contractuelles en fonction d'événements extérieurs. En général, ils fournissent une signature spécifique qui correspond au résultat d'un événement. Cette signature est ensuite utilisée pour compléter et rendre valide une transaction d'exécution qui envoie les bitcoins à la partie qui est censée les recevoir selon les conditions du contrat intelligent.

Terme associé : **DLC - DISCREET LOG CONTRACT**

ORDINAL NUMBER

COUCHE SUPÉRIEURE | FR : NOMBRE ORDINAL

Dans le cadre du protocole Ordinals, c'est un identifiant unique attribué à chaque sat en fonction de son ordre de minage. Ces numéros permettent de rendre non fongibles ces sats selon le protocole Ordinals, et donc de suivre et de transférer ces sats spécifiques.

Termes associés : **DIGITAL ARTIFACTS** **INSCRIPTIONS** **ORDINALS THEORY**

ORDINALS THEORY

COUCHE SUPÉRIEURE | FR : THÉORIE DES ORDINALS

Protocole externe à Bitcoin qui attribue des numéros de série aux sats (la plus petite unité de bitcoin), qui permettent de les tracer de manière individuelle et de les transférer via des transactions Bitcoin. Selon la théorie des Ordinals, chaque sat est numéroté selon l'ordre dans lequel il a été miné et est transféré de manière FIFO (*First-In-First-Out*). L'objectif de cette théorie est de rendre des sats non fongibles selon leur interprétation au sein du protocole Ordinals, afin de leur associer des informations comme des images (NFT) que l'on appelle des « inscriptions ».

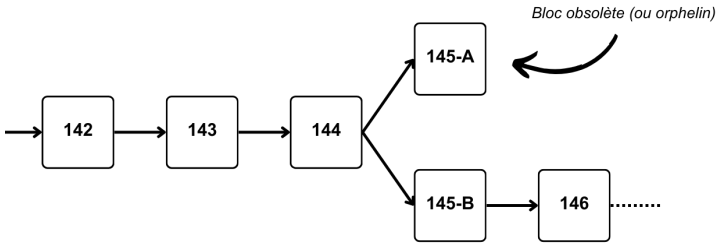
Termes associés : **DIGITAL ARTIFACTS** | **INSCRIPTIONS**

ORPHELIN

PROTOCOLE | EN : ORPHAN BLOCK

Théoriquement, un bloc orphelin désigne un bloc valide réceptionné par un nœud qui n'a pas encore acquis le bloc parent, c'est-à-dire le précédent dans la chaîne. Ce bloc, bien que valide, demeure isolé localement en tant qu'orphelin.

Cependant, dans l'usage courant, l'expression « bloc orphelin » fait souvent référence à un bloc sans enfant : un bloc valide, mais non retenu dans la chaîne principale de Bitcoin. Il se produit lorsque deux mineurs trouvent un bloc valide sur une même hauteur de chaîne durant un court laps de temps et le diffusent sur le réseau. Les nœuds finissent par choisir un seul bloc à inclure dans la chaîne, selon le principe de la chaîne avec le plus de travail accumulé, rendant l'autre « orphelin ».



Personnellement, je préfère employer le terme de « bloc orphelin » pour parler d'un bloc sans parent et le terme de « bloc obsolète » (stale block) pour désigner un bloc qui n'a pas d'enfant. Je trouve cela plus logique et compréhensible, bien qu'une majorité de bitcoiners ne suivent pas cet usage.

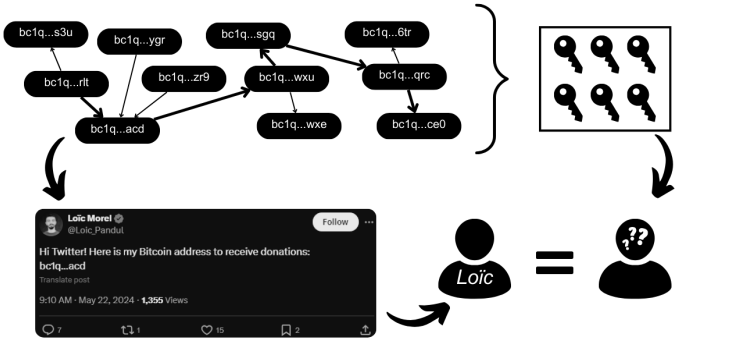
Termes associés : **BLOC** | **MINEUR**

OSINT

</> INFORMATIQUE

Acronyme de « *Open Source Intelligence* ». L'OSINT désigne la collecte et l'analyse d'informations disponibles publiquement à partir de sources accessibles à tous. Ces sources peuvent inclure des sites web, des forums, des réseaux sociaux, des bases de données publiques, des publications académiques, des documents gouvernementaux, etc. L'objectif principal de l'OSINT est de transformer des données brutes en informations exploitables, en identifiant des tendances, des corrélations et des pistes d'investigation.

Dans le cadre de Bitcoin, l'OSINT peut être utilisée dans le but d'appuyer une analyse de chaîne pour tracer des fonds, notamment afin d'identifier un point d'entrée, c'est-à-dire un lien entre une activité on-chain et une forme d'identité appartenant à une entité réelle. Par exemple, si vous publiez votre adresse de réception sur Twitter sous votre nom, un analyste pourrait la retrouver et l'associer à votre identité.



Termes associés : **ANALYSE DE CHAINE** | **POINT D'ENTRÉE**

OUTBOUND CAPACITY

LIGHTNING NETWORK | FR : CAPACITÉ SORTANTE

Désigne la quantité maximale de bitcoins qu'un nœud peut envoyer à travers un canal spécifique sur le Lightning Network. Elle dépend des fonds que le nœud a engagés dans le canal lors de son ouverture, ou qu'il a reçus lors d'un paiement Lightning ou d'un routage.

Termes associés : **INBOUND CAPACITY** | **LIGHTNING NETWORK**

OUTPOINT

PROTOCOLE | FR : POINT DE SORTIE

Référence unique à une sortie de transaction (*output*). Il est constitué de deux éléments :

- `txid` : l'identifiant de la transaction qui a créé l'output ;
- `vout` : l'index de l'output dans la transaction.

La combinaison de ces deux éléments permet d'identifier précisément un output

de transaction. Par exemple, si une transaction a un `txid` de `abc...123` et que l'index de l'output est `0`, l'outpoint sera noté :

```
abc...123:0
```

L'outpoint est utilisé dans les inputs (`vin`) d'une nouvelle transaction pour indiquer quel UTXO est dépensé.

Le terme « outpoint » est souvent utilisé comme synonyme de « UTXO ».

Termes associés : **INPUT** **UTXO**

OUTPUT

PROTOCOLE | FR : SORTIE

Dans le contexte de Bitcoin, un output au sein d'une transaction fait référence aux *Unspent Transaction Outputs* (UTXOs) qui sont créés comme fonds de destination pour le paiement. Plus précisément, il s'agit d'un mécanisme par lequel une transaction distribue des fonds. Une transaction prend des UTXOs, c'est-à-dire des morceaux de bitcoins, comme « inputs » (entrées) et crée de nouveaux UTXOs comme « outputs » (sorties). Ces outputs stipulent une certaine quantité de bitcoins, souvent attribués à une adresse spécifique, ainsi que les conditions sous lesquelles ces fonds peuvent être dépensés ultérieurement.

Le rôle de la transaction Bitcoin est donc de consommer des UTXOs en entrées, et de créer des nouveaux UTXOs en sorties. La différence entre les deux correspond aux frais de transactions qui peuvent être récupérés par le mineur gagnant du bloc. Un UTXO est, en essence, la sortie d'une transaction précédente qui n'a pas encore été dépensée. Les outputs de transaction sont donc les créations de nouveaux UTXOs qui seront, à leur tour, potentiellement utilisés comme inputs dans les transactions futures.

D'un point de vue plus large, en informatique, le terme « output » désigne généralement les données en résultat d'une fonction, d'un algorithme, ou d'un système. Par exemple, lorsque l'on passe une donnée dans une fonction de hachage cryptographique, cette information est nommée « input », et le résultat est nommé « output ».

Termes associés : **INPUT** **UTXO**

OUTPUT LINKING

CONFIDENTIALITÉ | FR : LIAISON DE SORTIES

Synonyme parfois utilisé pour parler de réutilisation d'adresse. L'*output linking* se réfère à la pratique d'utiliser une même adresse de réception pour bloquer plusieurs UTXOs, parfois au sein de plusieurs transactions différentes. Les bitcoins sont généralement bloqués à l'aide d'une paire de clés cryptographique qui correspond à une adresse unique. Puisque la blockchain est publique, il est facile de pouvoir consulter quelles adresses sont associées à combien de bitcoins. En cas de réutilisation d'une même adresse pour plusieurs paiements, on peut imaginer que tous les UTXOs associés appartiennent à une même entité. La réutilisation d'adresse pose donc un problème pour la vie privée de l'utilisateur. Elle permet de faire des liens déterministes entre plusieurs transactions

et plusieurs UTXOs, ainsi que de perpétuer un traçage de fonds *on-chain*.

Terme associé : **RÉUTILISATION D'ADRESSE**

OUTPUT SCRIPT DESCRIPTORS

PORTEFEUILLE

Les output script descriptors, ou simplement descriptors, sont des expressions structurées qui décrivent intégralement un script de sortie (`scriptPubKey`) et fournissent toutes les informations nécessaires pour suivre les transactions vers ou depuis un script particulier. Ces descriptors facilitent la gestion des clés dans les portefeuilles HD grâce à une description standard de la structure et des types d'adresses utilisés.

L'intérêt principal des descriptors réside dans leur capacité à encapsuler toutes les informations essentielles à la restauration d'un portefeuille dans une unique chaîne de caractères (en plus de la phrase de récupération). En sauvegardant un descriptor avec les phrases mnémoniques correspondantes, il est possible de restaurer non seulement les clés privées, mais aussi la structure précise du portefeuille et les paramètres de script associés. En effet, la récupération d'un portefeuille requiert non seulement la connaissance de la graine initiale, mais aussi des index spécifiques pour la dérivation des paires de clés enfants, ainsi que des `xpub` de chaque facteur dans le cadre d'un portefeuille multi-sig. Autrefois, on présumait que ces informations étaient implicitement sues de tous. Cependant, avec la diversification des scripts et l'émergence de configurations plus complexes, ces informations pourraient devenir difficiles à extrapoler, transformant ainsi ces données en informations privées et difficilement bruteforçables. L'utilisation de descriptors simplifie grandement le processus : il suffit de connaître la ou les phrases de récupération et le descriptor correspondant pour tout restaurer de façon fiable et sécurisée.

Un descriptor se compose de plusieurs éléments :

- Des fonctions de script comme `pk` (Pay-to-PubKey), `pkh` (Pay-to-PubKey-Hash), `wpkh` (Pay-to-Witness-PubKey-Hash), `sh` (Pay-to-Script-Hash), `wsh` (Pay-to-Witness-Script-Hash), `tr` (Pay-to-Taproot), `multi` (Multisignature) et `sortedmulti` (Multisignature avec clés triées) ;
- Des chemins de dérivation, par exemple `[d34db33f/44h/0h/0h]` qui indique un chemin dérivé et une empreinte de clé maîtresse spécifique ;
- Des clés en divers formats tels que des clés publiques en hexadécimal ou des clés publiques étendues (`xpub`) ;
- Une somme de contrôle, précédée d'un dièse, pour vérifier l'intégrité du descriptor.

Par exemple, un descriptor pour un portefeuille P2WPKH pourrait ressembler à :

```
wpkh([cdeab12f/84h/0h/0h]xpub6CUGRUonZSQ4TWtTMmzXdrXDtyPWKiKbERr
4d5qkSmh5h17C1TjvMt7DJ9Qve4dRxm91CDv6cNfKsq2mK1rMsJKhtRUPZz7MQtp
3y6atC1U/<0;1>/*)#jy0l7nr4
```

Dans ce descriptor, la fonction de dérivation `wpkh` indique un type de script

Pay-to-Witness-Public-Key-Hash. Elle est suivie par le chemin de dérivation qui contient :

- `cdeab12f` : l'empreinte de la clé maîtresse ;
- `84h` : qui signifie l'utilisation d'un objectif BIP-0084, destiné aux adresses SegWit v0 ;
- `0h` : qui indique qu'il s'agit d'une devise BTC sur le mainnet ;
- `0h` : qui fait référence au numéro de compte spécifique utilisé dans le portefeuille.

Le descriptor inclut également la clé publique étendue utilisée sur ce portefeuille :

```
xpub6CUGRUonZSQ4TWtTMmzXdrXDtyPWKiKbERr4d5qkSmh5h17C1TjvMt7DJ9Qv
e4dRxm91CDv6cNfKsq2mK1rMsJKhtRUPZz7MQtp3y6atC1U
```

Ensuite, la notation `/<0;1>/*` spécifie que le descriptor peut générer des adresses à partir de la chaîne externe (`0`) et interne (`1`), avec un wildcard (`*`) permettant la dérivation séquentielle de plusieurs adresses de manière paramétrable, similaire à la gestion d'un « gap limit » sur des logiciels de portefeuille classiques.

Enfin, `#jy0l7nr4` représente la somme de contrôle pour vérifier l'intégrité du descriptor.

Termes associés : **CHEMIN DE DÉRIVATION** | **SCRIPTPUBKEY**

OVERCLOCKING

↗ MINAGE

Pratique qui consiste à augmenter la fréquence de fonctionnement des puces ASICs utilisées pour le minage au-delà de leurs spécifications nominales. Cela permet de produire plus de hashrate et d'augmenter les chances de trouver un bloc. Cependant, cette pratique augmente également la consommation d'électricité des machines, et donc génère plus de chaleur, ce qui peut entraîner une usure prématurée des composants et nécessiter des systèmes de refroidissement améliorés, comme le refroidissement par immersion.

Termes associés : **ASIC** | **HASHRATE**

OVERT ASICBOOST

↗ MINAGE

Version ouverte et transparente d'AsicBoost. AsicBoost est une technique d'optimisation algorithmique utilisée dans le minage de Bitcoin. Les mineurs utilisant la version Overt manipulent le champ `nVersion` du bloc candidat et utilisent cette modification comme un nonce supplémentaire. Cette modification du champ `nVersion` permet de changer les données du premier chunk SHA256 de l'en-tête, et ainsi de réutiliser le résultat intermédiaire (*midstate*) de ce premier chunk pour toutes les itérations du `Nonce`, ce qui réduit les calculs nécessaires. Cette version est détectable publiquement et ne dissimule pas ses modifications au reste du réseau, à l'inverse de la version Covert d'AsicBoost.

Termes associés : **ASICBOOST** | **COVERT ASICBOOST**

OWNED STATE

RGB

Dans le cadre du protocole RGB, l'Owned State désigne la portion du Contract State encapsulée dans un Assignment et explicitement associée à un détenteur particulier par le biais d'un Single-use Seal pointant vers un UTXO. Cette composante représente, par exemple, un actif numérique ou un droit contractuel spécifique attribué à un individu. En isolant l'état lié à chaque titulaire, l'Owned State garantit une gestion précise et sécurisée de la propriété au sein des smart contracts RGB.

Termes associés : **CONTRACT STATE** | **SINGLE-USE SEAL**

OWNERSHIP

RGB

Dans le cadre du protocole RGB, le terme Ownership désigne la propriété qui, à travers la Seal Definition pointant vers un UTXO Bitcoin, définit qui peut modifier l'état d'un contrat. Lorsqu'un Owned State est associé à un UTXO, le propriétaire de ce dernier détient le droit, conformément aux règles du contrat, de transférer ou de faire évoluer l'état correspondant.

Termes associés : **OWNED STATE** | **SEAL DEFINITION**

OXT

CONFIDENTIALITÉ

Outil d'analyse exploratoire de la blockchain Bitcoin, créé par LaurentMT puis acquis par Samourai Wallet. OXT proposait des fonctionnalités avancées d'exploration et de visualisation de la chaîne, avec une orientation particulière vers l'étude de la confidentialité des transactions.

La plateforme permettait de visualiser les graphes de transactions, de calculer des métriques de confidentialité comme le score de Boltzmann et d'identifier des *patterns* d'utilisation (*coinjoins*, consolidations, etc.). Elle servait ainsi à comprendre et à auditer le niveau de vie privée offert par une transaction ou un ensemble de transactions.

Le service, autrefois accessible gratuitement à l'adresse oxt.me, est hors ligne depuis avril 2024, à la suite des poursuites judiciaires engagées contre les fondateurs de Samourai Wallet.

Terme associé : **COINJOIN**

P

P2MS

SCRIPT

P2MS est le sigle pour *Pay to Multisig* (en français « payer aux multiples signatures »). C'est un modèle de script standard utilisé pour établir des conditions de dépense sur un UTXO. Il permet de bloquer des bitcoins à l'aide de plusieurs clés publiques. Pour dépenser ces bitcoins, il faut fournir une signature avec un nombre prédéfini de clés privées associées. Par exemple, un `P2MS 2/3` dispose de `3` clés publiques avec `3` clés privées secrètes associées. Pour dépenser les bitcoins bloqués avec ce script P2MS, il faut réaliser une signature avec au moins `2` parmi les `3` clés privées. C'est un système de sécurisation à seuil (*threshold*).

Ce script a été inventé en 2011 par Gavin Andresen alors qu'il venait de récupérer la maintenance du client principal de Bitcoin. Aujourd'hui, le P2MS n'est utilisé qu'à la marge par certaines applications. L'extrême majorité des multisignatures modernes emploient d'autres scripts comme le P2SH ou le P2WSH. Par rapport à ceux-ci, le P2MS est extrêmement trivial. Les clés publiques le constituant sont dévoilées dès la réception de la transaction. L'utilisation d'un P2MS est également plus chère que les autres scripts multisignatures.

Les P2MS sont souvent nommés « bare-multisig », ce qui peut être traduit en français par « multisignatures nues », ou « multisignatures brutes ». Au début de l'année 2023, les scripts P2MS étaient au centre d'une polémique à cause de leur utilisation détournée au sein du protocole Stamps. Leur impact sur l'UTXO set était notamment pointé du doigt.

Termes associés : **BARE-MULTISIG** | **P2SH**

P2P - PAIR-À-PAIR

PROTOCOLE | EN : P2P - PEER-TO-PEER

Fait référence à un modèle de communication et de distribution de données dans lequel les participants, souvent appelés nœuds ou pairs, partagent leurs ressources (comme des fichiers, de la puissance de traitement, de la bande passante, des actifs…) directement entre eux, sans nécessiter d'intermédiaire centralisé. Dans un système P2P, chaque participant agit simultanément comme client (consommateur de ressources) et serveur (fournisseur de ressources).

Le système Bitcoin fonctionne selon un modèle P2P, où les nœuds sont responsables de la validation des transactions et de la conservation de la blockchain. Cela signifie que, contrairement aux systèmes bancaires traditionnels qui dépendent d'entités centralisées, Bitcoin opère sur une structure distribuée où aucune entité unique ne détient le contrôle. Les nœuds du réseau Bitcoin communiquent entre eux pour diffuser les transactions et les blocs, et trouver un consensus sur l'état du registre.

Termes associés : **CONSENSUS** | **NOEUD**

P2P TRANSPORT V2

RÉSEAU

Nouvelle version du protocole de transport Bitcoin P2P intégrant le chiffrement opportuniste pour améliorer la confidentialité et la sécurité des communications entre les nœuds. Cette amélioration vise à résoudre plusieurs problématiques de la version de base du protocole P2P, notamment en rendant les données échangées indiscernables pour un observateur passif (tel qu'un fournisseur d'accès à internet), réduisant ainsi les risques de censure et d'attaques par détection de motifs spécifiques dans les paquets de données.

Le chiffrement mis en place n'inclut pas d'authentification afin de ne pas ajouter de complexité inutile, et de ne pas compromettre le fait que la connexion au réseau reste sans permission. Ce nouveau protocole de transport P2P offre néanmoins une meilleure sécurité contre les attaques passives et rend les attaques actives nettement plus coûteuses et détectables (notamment les attaques MITM). L'introduction d'un flux de données pseudo-aléatoire complique la tâche des attaquants souhaitant censurer ou manipuler les communications.

Le transport P2P V2 a été inclus en option (désactivé par défaut) dans la version 26.0 de Bitcoin Core, déployée en décembre 2023. Il peut être activé avec l'option `v2transport=1` dans le fichier de configuration.

Termes associés : **BIP-0324** | **CHIFFRER - CHIFFREMENT**

P2PK

SCRIPT

P2PK est le sigle pour *Pay to Public Key* (en français « payer à une clé publique »). C'est un modèle de script standard utilisé sur Bitcoin pour établir des conditions de dépense sur un UTXO. Il permet de bloquer des bitcoins directement sur une clé publique, plutôt que sur une adresse. Techniquement, le script P2PK contient une clé publique et une instruction qui exige une signature numérique correspondante pour débloquer les fonds. Lorsque le propriétaire souhaite dépenser les bitcoins, il doit fournir une signature produite avec la clé privée associée. Cette signature est vérifiée avec l'algorithme ECDSA (*Elliptic Curve Digital Signature Algorithm*). P2PK était souvent utilisé dans les premières versions de Bitcoin, notamment par Satoshi Nakamoto. Il n'est presque plus utilisé à ce jour.

Termes associés : **ECDSA** | **P2PKH**

P2PKH

SCRIPT

P2PKH est le sigle pour *Pay to Public Key Hash* (en français « payer au hachage d'une clé publique »). C'est un modèle de script standard utilisé pour établir des conditions de dépense sur un UTXO. Il permet de bloquer des bitcoins sur un hachage d'une clé publique, c'est-à-dire sur une adresse de réception. Ce script est associé au standard Legacy, et a été introduit dès les premières versions de Bitcoin par Satoshi Nakamoto.

À la différence du P2PK, où la clé publique est explicitement incluse dans le script, le P2PKH fait appel à une empreinte cryptographique de la clé publique. Ce script inclut le hachage `RIPEMD160` du `SHA256` de la clé publique et stipule que, pour accéder aux fonds, le destinataire doit fournir une clé publique correspondant à ce hachage, ainsi qu'une signature numérique valide générée à partir de la clé privée associée. Les adresses P2PKH sont encodées en utilisant le format `Base58Check`, ce qui leur confère une robustesse contre les erreurs typographiques grâce à l'utilisation d'une somme de contrôle. Ces adresses débutent systématiquement par le chiffre `1`.

Termes associés : **BASE58CHECK** **P2PK**

P2POOL

MINAGE

Protocole de pool de minage décentralisé pour Bitcoin, conçu pour éliminer le besoin d'un opérateur central. P2Pool fonctionne en créant une blockchain secondaire appelée « share chain », dont la difficulté est ajustée pour produire un bloc environ toutes les 30 secondes. Les blocs de cette *share chain* sont structurellement identiques à ceux de la blockchain Bitcoin, mais avec une cible de difficulté plus faible. Chaque mineur exploite son propre nœud P2Pool connecté à un nœud Bitcoin complet et contribue au réseau pair-à-pair de la *share chain*.

Lorsqu'un des blocs de la *share chain* satisfait également la difficulté requise par le réseau Bitcoin, il est diffusé sur le réseau Bitcoin. La récompense du bloc est alors distribuée parmi les mineurs ayant contribué les shares les plus récentes dans la *share chain*. Ce mécanisme garantit que la distribution des récompenses est proportionnelle à la contribution de chaque mineur, sans qu'un opérateur central ne contrôle les fonds ou la sélection des transactions.

P2Pool a été créé en 2011 par Forrest Voight. Il a connu une adoption significative dans les premières années de Bitcoin, mais son utilisation a décliné face à la croissance des pools centralisées, qui offraient des interfaces plus simples et une variance plus faible pour les mineurs individuels. Le dernier bloc miné par P2Pool date de 2019.

Termes associés : **MINAGE** **MINING POOL**

P2SH

SCRIPT

P2SH est le sigle pour *Pay to Script Hash* (en français « payer au hachage du script »). C'est un modèle de script standard utilisé pour établir des conditions de dépense sur un UTXO. Contrairement aux scripts P2PK et P2PKH, où les conditions de dépense sont prédéfinies, P2SH permet l'intégration de conditions de dépense arbitraires et de fonctionnalités additionnelles au sein d'un script de transaction.

Techniquement, dans une transaction P2SH, le `scriptPubKey` contient l'empreinte cryptographique d'un `redeemScript`, plutôt que de conditions de dépense explicites. Cette empreinte est obtenue en utilisant un hachage `HASH160`, c'est-à-dire un `SHA256` suivi d'un `RIPEMD160`. Lors de l'envoi de bitcoins à une adresse P2SH, le `redeemScript` sous-jacent n'est pas révélé. Seule son empreinte est incluse dans la transaction. Les adresses P2SH sont encodées en `Base58Check` et commencent par le chiffre `3`. Lorsque le destinataire souhaite dépenser les bitcoins reçus, il doit fournir un `redeemScript` correspondant à l'empreinte présente dans le `scriptPubKey`, ainsi que les données nécessaires pour satisfaire les conditions de ce `redeemScript`. Par exemple, dans un P2SH multisignatures, le script pourrait exiger des signatures de plusieurs clés privées.

L'utilisation de P2SH donne plus de flexibilité, car il permet la construction de scripts arbitraires sans que l'expéditeur en connaisse les détails. P2SH est introduit en 2012 avec le BIP-0016.

Termes associés : **BIP-0016** **REDEEMSCRIPT**

P2SH-P2WPKH

SCRIPT

P2SH-P2WPKH est le sigle pour *Pay to Script Hash - Pay to Witness Public Key Hash* (en français « payer au hachage du script - payer au témoin du hachage de la clé publique »). C'est un modèle de script standard utilisé pour établir des conditions de dépense sur un UTXO, également connu sous le nom de « Nested SegWit ».

P2SH-P2WPKH a été introduit avec l'implémentation de SegWit en août 2017. Ce script est un P2WPKH enveloppé au sein d'un P2SH. Il verrouille des bitcoins sur la base du hachage d'une clé publique. La différence avec le P2WPKH classique est que le script est ici enveloppé dans le `redeemScript` d'un P2SH classique.

Ce script a été créé au lancement de SegWit pour faciliter son adoption. Il permet d'utiliser ce nouveau standard, même avec des services ou des *wallets* pas encore compatibles nativement avec SegWit. C'est une sorte de script de transition vers la nouvelle norme. Aujourd'hui, il n'est donc plus très pertinent d'utiliser ce type de scripts SegWit *wrappés*, puisque la plupart des *wallets* ont implémenté du SegWit natif. Les adresses P2SH-P2WPKH sont écrites en utilisant l'encodage `Base58Check` et commencent toujours par `3`, comme n'importe quelle adresse P2SH.

`P2SH-P2WPKH` *est également parfois appelé* `P2WPKH-nested-in-P2SH`.

Termes associés : **NESTED SEGWIT** | **P2WPKH**

P2SH-P2WSH

SCRIPT

P2SH-P2WSH est le sigle pour *Pay to Script Hash - Pay to Witness Script Hash* (en français « payer au hachage du script - payer au témoin du hachage du script »). C'est un modèle de script standard utilisé pour établir des conditions de dépense sur un UTXO, également connu sous le nom de « *Nested SegWit* ».

P2SH-P2WSH a été introduit avec l'implémentation de SegWit en août 2017. Ce script décrit un P2WSH enveloppé au sein d'un P2SH. Il verrouille des bitcoins sur la base du hachage d'un script. La différence avec le P2WSH classique est que le script est enveloppé dans le `redeemScript` d'un P2SH classique.

Ce script a été créé au lancement de SegWit pour faciliter son adoption. Il permet d'utiliser ce nouveau standard, même avec des services ou des *wallets* pas encore compatibles nativement avec SegWit. Aujourd'hui, il n'est donc plus très pertinent d'utiliser ce type de scripts SegWit *wrappés*, puisque la plupart des *wallets* ont implémenté du SegWit natif. Les adresses P2SH-P2WSH sont écrites en utilisant l'encodage `Base58Check` et commencent toujours par `3`, comme n'importe quelle adresse P2SH.

Termes associés : **NESTED SEGWIT** | **P2WSH**

P2TR

SCRIPT

P2TR est le sigle pour *Pay to Taproot* (en français « payer à la racine »). C'est un modèle de script standard utilisé pour établir des conditions de dépense sur un UTXO. Il a été introduit avec l'implémentation de Taproot en novembre 2021. P2TR utilise le protocole de Schnorr pour agréger des clés cryptographiques, ainsi que des arbres de Merkle pour des scripts alternatifs, connus sous le nom de MAST (*Merkelized Alternative Script Tree*). Contrairement aux transactions traditionnelles où les conditions de dépense sont exposées publiquement (parfois à la réception, parfois à la dépense), P2TR permet de masquer des scripts complexes derrière une seule clé publique apparente.

Techniquement, un script P2TR verrouille des bitcoins sur une clé publique Schnorr unique, dénommée K. Cependant, cette clé K est en réalité un agrégat d'une clé publique P et d'une clé publique M, cette dernière étant calculée à partir de la racine de Merkle d'une liste de `scriptPubKey`. Les bitcoins verrouillés avec un script P2TR peuvent être dépensés de deux manières distinctes : soit en publiant une signature pour la clé publique P, soit en satisfaisant l'un des scripts contenus dans l'arbre de Merkle. La première option est appelée « *key path* » (chemin de clé) et la seconde « *script path* » (chemin de script).

Ainsi, P2TR permet aux utilisateurs d'envoyer des bitcoins soit à une clé publique, soit à plusieurs scripts de leur choix. Un autre avantage de ce script est que, bien qu'il y ait de multiples façons de dépenser une sortie P2TR, seule

celle qui est utilisée doit être révélée à la dépense, permettant ainsi aux alternatives inutilisées de rester privées. P2TR est une sortie SegWit de version 1, ce qui signifie que les signatures pour les entrées P2TR sont stockées dans le témoin d'une transaction, et non dans le `scriptSig`. Les adresses P2TR utilisent un encodage `Bech32m` et commencent par `bc1p`.

Termes associés : **MAST** **SCHNORR**

P2WPKH

SCRIPT

P2WPKH est le sigle pour *Pay to Witness Public Key Hash* (en français « payer au témoin du hachage de la clé publique »). C'est un modèle de script standard utilisé pour établir des conditions de dépense sur un UTXO. P2WPKH a été introduit avec l'implémentation de SegWit en août 2017.

Ce script est similaire à P2PKH (*Pay to Public Key Hash*), puisqu'il verrouille également des bitcoins sur la base du hachage d'une clé publique, c'est-à-dire d'une adresse de réception. La différence réside dans la manière dont les signatures et les scripts sont inclus dans la transaction. Dans le cadre de P2WPKH, les informations du script de signature (`scriptSig`) sont déplacées de la structure traditionnelle de la transaction vers une section distincte appelée `Witness` (témoin). Ce déplacement est une caractéristique de la mise à jour SegWit (*Segregated Witness*) qui permet d'empêcher la malléabilité liée à la signature. Les transactions P2WPKH sont généralement moins coûteuses en termes de frais par rapport aux transactions Legacy, car la partie dans le témoin coûte moins cher.

Les adresses P2WPKH sont écrites en utilisant l'encodage `Bech32` avec une somme de contrôle sous forme de code BCH. Ces adresses commencent toujours par `bc1q`. P2WPKH est une sortie SegWit de version 0.

Termes associés : **P2PKH** **SEGWIT**

P2WSH

SCRIPT

P2WSH est le sigle pour *Pay to Witness Script Hash* (en français « payer au témoin du hachage du script »). C'est un modèle de script standard utilisé pour établir des conditions de dépense sur un UTXO. P2WSH a été introduit avec l'implémentation de SegWit en août 2017.

Ce script est similaire à P2SH (*Pay to Script Hash*), puisqu'il verrouille également des bitcoins sur la base du hachage d'un script. La différence réside dans la manière dont les signatures et les scripts sont inclus dans la transaction. Pour dépenser les bitcoins sur ce type de script, le bénéficiaire doit fournir le script d'origine, appelé `witnessScript` (équivalent du `redeemScript`), ainsi que les signatures requises. Ce mécanisme permet d'implémenter des conditions de dépense plus sophistiquées, telles que des multisigs.

Dans le cadre de P2WSH, les informations du script de signature (le `scriptWitness`, l'équivalent du `scriptSig`) sont déplacées de la structure traditionnelle de la transaction vers une section distincte appelée `Witness`

(témoin). Ce déplacement est une caractéristique de la mise à jour SegWit (*Segregated Witness*) qui permet d'empêcher la malléabilité liée à la signature. Les transactions P2WSH sont généralement moins coûteuses en termes de frais par rapport aux transactions Legacy, car la partie dans le témoin coûte moins cher.

Les adresses P2WSH sont écrites en utilisant l'encodage `Bech32` avec une somme de contrôle sous forme de code BCH. Ces adresses commencent toujours par `bc1q`. P2WSH est une sortie SegWit de version 0.

Termes associés : **P2SH** | **SEGWIT**

PACKAGE RELAY

RÉSEAU | FR : RELAIS DE PAQUETS

Amélioration du protocole de relais de Bitcoin qui permet aux nœuds de transmettre et d'évaluer des groupes de transactions interdépendantes de manière collective, plutôt que de les traiter individuellement. Sans ce mécanisme, une transaction parente dont le taux de frais est trop faible pour être acceptée dans la mempool est rejetée, ce qui empêche toute transaction enfant de servir au CPFP (*Child Pays For Parent*), puisque la parente est nécessaire à sa validation.

Ce problème est particulièrement critique pour les protocoles à contrainte temporelle comme le Lightning Network, où les transactions sont signées bien avant leur diffusion. Le taux de frais minimal accepté par un nœud dépend du contenu de sa mempool à un instant donné ; une transaction parente auparavant éligible au CPFP peut donc ne plus l'être au moment de sa diffusion.

Le *package relay* résout ce problème en évaluant le taux de frais combiné d'un ensemble parent-enfant. L'implémentation actuelle dans Bitcoin Core se concentre sur une variante limitée à un parent et un enfant (1p1c), qui ne nécessite pas de modification du protocole P2P. Le BIP-0331 formalise la spécification plus générale du relais de paquets ancestraux. Le principal défi de développement réside dans la prévention des vulnérabilités de déni de service.

Ce mécanisme est complémentaire des *anchor outputs* et constitue une contremesure importante face aux attaques d'épinglage de transactions (*transaction pinning*).

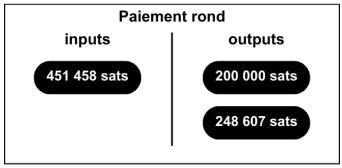
Termes associés : **CPFP - CHILD PAY FOR PARENT** | **GOSSIP** | **TRANSACTION PINNING**

PAIEMENT ROND

CONFIDENTIALITÉ | EN : ROUND PAYMENT

Heuristique interne d'analyse de chaîne sur Bitcoin qui permet d'émettre une hypothèse sur la nature des sorties d'une transaction en se basant sur les montants ronds. De manière générale, lorsque l'on se retrouve face à un *pattern* de paiement simple (1 ou plusieurs inputs et 2 outputs), si une des sorties dépense un montant rond, alors celle-ci représente le paiement. Par élimination, si une sortie représente le paiement, l'autre représente le change. On peut donc interpréter qu'il est vraisemblable que l'utilisateur en entrée soit toujours en possession de la sortie identifiée comme étant le change.

Il convient de souligner que cette heuristique n'est pas toujours applicable, puisque la majorité des paiements s'effectuent encore en unités de compte fiat. En effet, lorsqu'un commerçant en France accepte le bitcoin, en général, il n'affiche pas des prix stables en sats. Il optera plutôt pour une conversion entre le prix en euros et le montant en bitcoins à régler grâce à son POS (comme BTCPay Server par exemple). Il ne devrait donc pas y avoir de nombre rond en sortie de la transaction. Néanmoins, un analyste pourrait tenter de réaliser cette conversion en tenant compte du taux de change en vigueur lorsque la transaction a été diffusée sur le réseau. Si un jour, le bitcoin devient l'unité de compte préférée dans nos échanges, cette heuristique pourrait devenir encore plus utile pour les analyses.

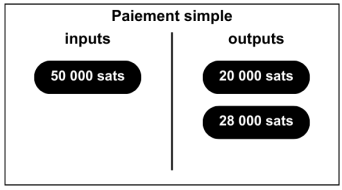


Termes associés : **ANALYSE DE CHAINE** | **PAIEMENT SIMPLE**

PAIEMENT SIMPLE

CONFIDENTIALITÉ | EN : SIMPLE PAYMENT

Pattern (ou modèle) de transaction utilisé en analyse de chaîne qui se caractérise par la consommation d'un ou plusieurs UTXOs en inputs et la production de 2 UTXOs en outputs. Ce modèle va donc ressembler à cela :



Ce modèle du paiement simple indique que nous sommes vraisemblablement en présence d'une transaction d'envoi ou de paiement. L'utilisateur a consommé son propre UTXO en inputs pour satisfaire en outputs un UTXO de paiement

et un UTXO de change (rendu de monnaie qui revient vers le même utilisateur). Nous savons donc que l'utilisateur observé n'est vraisemblablement plus en possession d'un des deux UTXOs en outputs (celui du paiement), mais qu'il est toujours en possession de l'autre UTXO (celui de change).

Termes associés : **ANALYSE DE CHAINE** | **PAIEMENT ROND**

PAIR ENTRANT

RÉSEAU | EN : INBOUND PEER / INCOMING CONNECTION

Nœud du réseau Bitcoin qui lance une connexion vers votre nœud sans intervention de votre part. Bitcoin Core autorise au maximum 114 pairs entrants par défaut (sur un total de 125 connexions), afin de faciliter la connectivité au sein du réseau. Les pairs entrants sont considérés avec prudence, car on ne peut pas être sûr qu'ils sont honnêtes, du fait qu'ils soient initiés par des tiers. Les pairs entrants et sortants partagent le même type d'informations. La principale différence entre les pairs entrants et sortants réside non pas dans le type d'informations échangées, mais dans la manière dont ces connexions sont établies.

Termes associés : **NOEUD** | **PAIR SORTANT**

PAIR SORTANT

RÉSEAU | EN : OUTBOUND PEER / OUTGOING CONNECTION

Nœud vers lequel votre propre nœud Bitcoin établit activement une connexion. Par défaut, un nœud tente de se connecter à 8 pairs sortants. Ces connexions sont privilégiées et considérées comme plus fiables que les pairs entrant, car elles sont choisies par le nœud. Les pairs entrants et sortants partagent le même type d'informations. La principale différence entre les pairs entrants et sortants réside non pas dans le type d'informations échangées, mais dans la manière dont ces connexions sont établies.

Termes associés : **NOEUD** | **PAIR ENTRANT**

PAIRING PHRASE

LIGHTNING NETWORK

Suite de mots utilisée par le protocole Lightning Node Connect (LNC) pour authentifier et sécuriser une connexion distante à un nœud Lightning. La *pairing phrase* est générée par le nœud via *litd* et transmise à l'utilisateur, qui la saisit dans Lightning Terminal ou toute autre application compatible. Elle établit un canal chiffré de bout en bout à travers un serveur proxy, sans nécessiter de VPN, de redirection de ports ou d'exposition directe du nœud sur internet.

Termes associés : **LIGHTNING TERMINAL** | **LNC - LIGHTNING NODE CONNECT**

PANNE BYZANTINE

ATTAQUE | EN : BYZANTINE FAULT

La panne byzantine, ou comportement byzantin, est tout comportement d'un système ne respectant pas ses propres spécifications, en donnant des résultats non conformes.

Les « pannes byzantines naturelles » sont couramment distinguées des « pannes byzantines volontaires » : les premières provenant généralement d'erreurs physiques non détectées (mémoire, transmissions réseaux, etc.), tandis que les secondes sont principalement issues d'attaques visant à faire échouer le système (sabotage, malveillance, porte dérobée).

L'authentification, la signature et le consensus par des moyens cryptographiques et des algorithmes « Tolérants aux pannes byzantines » ou « *Byzantine Fault Tolerant* (BFT) » permettent de limiter ces dernières.

La panne byzantine fait référence au problème des généraux byzantins, une métaphore traitant de la remise en cause de la fiabilité des transmissions et de l'intégrité des interlocuteurs.

Termes associés : **CONSENSUS** | **PROBLÈME DES GÉNÉRAUX BYZANTINS**

PAPER WALLET

PORTEFEUILLE | FR : PORTEFEUILLE PAPIER

Méthode de stockage hors ligne qui consiste à imprimer ou noter une paire de clés cryptographiques donnant accès à des bitcoins sur un papier (ou éventuellement un autre support physique) uniquement. Les portefeuilles papier sont considérés comme un moyen de stockage à froid, car ils ne sont pas connectés à internet, à condition qu'ils aient été produits manuellement et pas imprimés. Leur sécurité repose alors sur la protection physique du papier contre la perte, la destruction et le vol. Ils sont aujourd'hui souvent considérés comme obsolètes, car ils offrent un moins bon modèle de sécurité que les solutions modernes. De plus, leur utilisation implique souvent la réutilisation de la même clé publique, ce qui est très problématique pour la confidentialité de l'utilisateur.

Termes associés : **CLÉ PRIVÉE** | **COLD WALLET**

PASSIVE ASSETS

COUCHE SUPÉRIEURE | FR : ACTIFS PASSIFS

Dans le protocole Taproot Assets, les actifs passifs désignent les actifs que l'expéditeur conserve dans son arbre d'actifs (*asset tree*) lorsqu'il effectue un transfert. Lors d'un envoi, l'arbre est modifié : les actifs activement transférés vers un nouveau détenteur sont dits « actifs », tandis que ceux qui restent en possession de l'expéditeur sont qualifiés de « passifs ». Ces actifs passifs doivent être ré-ancrés (*re-anchored*) dans une nouvelle racine Merkle pour maintenir l'intégrité de l'arbre après la transaction.

Termes associés : **ASSET SPLIT** | **TAPROOT ASSETS PROTOCOL**

PASSPHRASE - BIP-0039

PORTEFEUILLE

Mot de passe optionnel qui, combiné à la phrase de récupération, offre une couche de sécurité supplémentaire pour les portefeuilles Bitcoin déterministes et hiérarchiques. Les portefeuilles HD sont généralement générés à partir d'une phrase de récupération constituée de 12 ou de 24 mots. Cette phrase de récupération est très importante, car elle permet de restaurer l'ensemble des clés d'un portefeuille en cas de perte. Cependant, elle constitue un point de défaillance unique (SPOF). Si elle est compromise, les bitcoins sont en danger. C'est ici qu'intervient la passphrase. C'est un mot de passe optionnel, choisi par l'utilisateur, qui s'ajoute à la phrase de récupération pour renforcer la sécurité du portefeuille. À ne pas confondre avec un code PIN ou un mot de passe ordinaire, la passphrase joue un rôle dans la dérivation des clés cryptographiques.

Elle fonctionne en tandem avec la phrase de récupération, modifiant la graine à partir de laquelle sont générées les clés. Ainsi, même si une personne obtient votre phrase de récupération, sans la passphrase, elle ne peut pas accéder à vos fonds. L'utilisation d'une passphrase crée essentiellement un nouveau portefeuille avec des clés distinctes. Modifier (même légèrement) la passphrase générera un portefeuille différent.

La passphrase est arbitraire et peut être n'importe quelle combinaison de caractères choisie par l'utilisateur. L'utilisation d'une passphrase offre plusieurs avantages. Tout d'abord, elle réduit les risques liés à la compromission de la phrase de récupération en nécessitant un second facteur pour accéder aux fonds. Ensuite, elle peut être utilisée stratégiquement pour créer des portefeuilles d'appât contenant de petites quantités de bitcoins, dans le cas d'une contrainte physique pour voler vos fonds. Enfin, son utilisation est intéressante lorsque l'on souhaite maitriser le caractère aléatoire de la génération de la graine du portefeuille HD. La passphrase doit être suffisamment complexe pour résister aux attaques par brute force et doit être sauvegardée de manière fiable. La perte de la passphrase peut entraîner l'incapacité d'accéder aux fonds, tout comme la perte de la phrase de récupération.

La passphrase est parfois également nommée : « two-factor seed phrase », « password », « seed extension », « extension word » ou encore « 13ème ou 25ème mot ». Notons qu'il existe deux types de passphrases sur Bitcoin. La plus connue est celle décrite ci-dessus, qui dépend du BIP-0039, et qui permet de sécuriser tout un portefeuille HD entier. Toutefois, le BIP-0038 avait également spécifié une manière de sécuriser une clé privée unique à l'aide d'une passphrase. Ce second type de passphrase n'est presque plus utilisé aujourd'hui.

Termes associés : **GRAINE** | **PHRASE DE RÉCUPÉRATION**

PASSPORT

PORTEFEUILLE

Hardware wallet fabriqué par la société américaine Foundation Devices, conçu exclusivement pour Bitcoin. Le Passport est un appareil entièrement *air-gapped* qui communique avec les logiciels de portefeuille via une caméra et des QR codes, sans aucune connexion filaire pour la signature des transactions. Il intègre un *secure element* et son *firmware* est *open source*. L'appareil dispose d'un écran couleur et d'un pavé directionnel, et est alimenté par une batterie lithium-ion amovible au format Nokia BL-5C. Le Passport fonctionne avec l'application compagnon Envoy, ainsi qu'avec des logiciels tiers comme Sparrow Wallet.

Termes associés : **HARDWARE WALLET** | **PSBT**

PATHFINDING

LIGHTNING NETWORK | FR : RECHERCHE DE CHEMIN

Processus utilisé par un nœud pour déterminer le chemin optimal afin de router un paiement à travers le réseau de canaux Lightning. Le *pathfinding* est réalisé par le nœud du payeur, qui doit sélectionner les nœuds intermédiaires les plus adaptés pour atteindre le destinataire. Ce choix est basé sur plusieurs critères tels que les frais, la capacité des canaux et les *timelocks*.

Les algorithmes de *pathfinding* construisent un graphe de la topologie du réseau à partir des données de *gossip* et évaluent les différentes routes possibles entre le nœud payeur et le receveur. Ces routes sont classées de la meilleure à la moins bonne selon divers critères. Le nœud teste ensuite ces routes jusqu'à réussir à effectuer le paiement sur l'une d'entre elles.

Termes associés : **CANAL DE PAIMENT** | **GOSSIP**

PATOSHI

HISTOIRE

Fait référence à un motif distinct de nonces et d'horodatages observés dans les blocs minés au cours des premiers mois de l'existence de Bitcoin. Ce modèle est soupçonné d'être attribuable à une seule entité ou individu, très probablement Satoshi Nakamoto lui-même, l'inventeur de Bitcoin. Le terme de « Patoshi » est un mot-valise combinant « *pattern* » et « Satoshi ». Plusieurs analyses des premiers blocs de Bitcoin ont révélé des motifs dans la façon dont les nonces et extra-nonces étaient choisis, et comment les horodatages étaient définis. Ces motifs étaient suffisamment distincts pour suggérer qu'un mineur ou un groupe de mineurs particulier était responsable d'une grande proportion des blocs minés pendant cette période, tout en utilisant un client modifié. Sergio Demian Lerner, un chercheur en informatique, est crédité de la découverte de ce motif en 2013. Lerner a estimé que le mineur Patoshi avait miné environ 1,1 million de bitcoins. Cela a conduit à des spéculations généralisées sur les motivations, l'identité et les intentions actuelles du mineur Patoshi. Certains pensent que Patoshi était Satoshi lui-même, en train de miner des bitcoins pour soutenir et sécuriser le réseau naissant. Il est important de noter que, bien que le motif Patoshi soit largement accepté comme preuve d'une activité de minage

précoce concentrée, il n'y a pas de confirmation définitive de l'identité derrière le Patoshi ou de ses intentions.

Termes associés : **MINAGE** **NAKAMOTO SATOSHI**

PAY-TO-CONTRACT

CRYPTOGRAPHIE

Protocole cryptographique qui permet d'engager des données arbitraires dans une clé publique Bitcoin, de sorte que le paiement effectué vers cette clé constitue simultanément une preuve d'engagement envers ces données.

Le principe repose sur le « tweaking » d'une clé publique : à partir d'une clé publique originale P et d'un contrat c, on calcule une nouvelle clé $P' = P + H(P\|c) \cdot G$, où H est une fonction de hachage et G le point générateur de la courbe. La clé P' qui en résulte semble être une clé publique ordinaire, mais elle encode un engagement vérifiable envers le contrat c. Le payeur peut ensuite prouver à un tiers ce pour quoi il a payé, en révélant la clé originale P et le contrat c, sans que cette preuve soit visible sur la blockchain.

Termes associés : **MULTISIG** **SCHNORR**

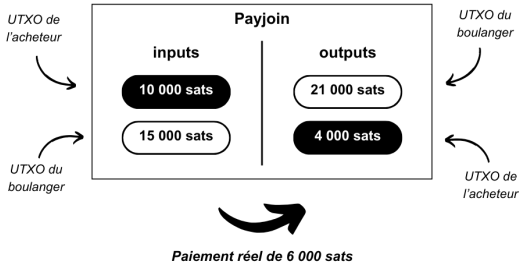
PAYJOIN

CONFIDENTIALITÉ

Structure spécifique de transaction Bitcoin qui permet d'améliorer la confidentialité des utilisateurs lors d'une dépense en collaborant avec le destinataire du paiement. La particularité du Payjoin réside dans sa capacité à générer une transaction qui paraît ordinaire à première vue, mais qui est en réalité un mini coinjoin entre deux personnes. Pour cela, la structure de la transaction fait intervenir le destinataire du paiement dans les entrées aux côtés de l'expéditeur réel. Le destinataire inclut donc un paiement vers lui-même au milieu de la transaction qui permet elle-même de le payer. Par exemple, si vous achetez une baguette pour `6 000 sats` à l'aide d'un UTXO de `10 000 sats`, et que vous optez pour un Payjoin, votre boulanger ajoutera un UTXO de `15 000 sats` lui appartenant en entrée, qu'il récupèrera en intégralité en sortie, en plus de vos `6 000 sats`.

La transaction Payjoin remplit deux objectifs. Tout d'abord, elle vise à induire en erreur un observateur extérieur en créant un leurre dans l'analyse de chaîne sur l'heuristique CIOH (*Common Input Ownership Heuristic*). Habituellement, lorsqu'une transaction sur la blockchain présente plusieurs entrées, on suppose que toutes ces entrées appartiennent vraisemblablement à une même entité. Ainsi, lorsqu'un analyste examine une transaction Payjoin, il est amené à croire que toutes les entrées proviennent d'une même personne. Toutefois, cette perception est erronée, car le destinataire du paiement contribue également aux entrées aux côtés du payeur réel. Ensuite, le Payjoin permet également de tromper un observateur extérieur sur le montant réel du paiement qui a été opéré. En examinant la structure de la transaction, l'analyste pourrait croire que le paiement est équivalent au montant d'une des sorties. En réalité, le montant du paiement ne correspond à aucun des outputs. Il est en fait la différence entre l'UTXO du destinataire en sortie et l'UTXO du destinataire en entrée. En cela, la

transaction Payjoin rentre dans le domaine de la stéganographie. Elle permet de cacher le montant réel d'une transaction au sein d'une fausse transaction qui agit comme un leurre.



Le Payjoin est également parfois nommé « P2EP (Pay-to-End-Point) », « Stowaway » ou « transaction stéganographique ».

Termes associés : **CIOH** | **STÉGANOGRAPHIE**

PAYMENT HASH

LIGHTNING NETWORK | FR : HACHAGE DE PAIEMENT

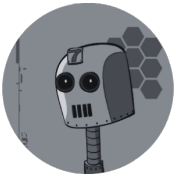
Empreinte cryptographique de la *preimage* utilisée pour sécuriser un paiement sur le Lightning Network. Lorsqu'un destinataire crée une *invoice*, il génère un secret aléatoire (la *preimage*), en calcule le hash et l'intègre dans l'*invoice*. L'expéditeur construit alors une chaîne de HTLC verrouillés par ce hash le long de la route de paiement. Le paiement n'est finalisé que lorsque le destinataire révèle la *preimage* correspondante, permettant à chaque nœud intermédiaire de débloquer ses fonds successivement.

Termes associés : **HTLC** | **PREIMAGE**

PAYNYM

CONFIDENTIALITÉ

Identifiant unique lié à un portefeuille Bitcoin qui implémente cette option. Les Paynyms sont disponibles sur Samourai Wallet et sur Sparrow Wallet. À l'origine, ces identifiants sont générés à partir du code de paiement du portefeuille, conformément au BIP-0047. Ils offrent ainsi la possibilité de se connecter avec un autre utilisateur afin de générer des adresses de réception vierges et uniques, et ce, sans nécessiter d'interaction directe. L'usage des Paynyms a ensuite été élargi pour soutenir diverses autres fonctionnalités de l'écosystème Samourai. Ils sont notamment employés pour permettre des échanges chiffrés sur le protocole de communication Soroban, afin de mettre en œuvre facilement des transactions collaboratives. Les Paynyms peuvent aussi servir à l'authentification sur des systèmes compatibles avec le protocole AUTH47. Chaque Paynym se caractérise par un identifiant unique et est représenté par une illustration de robot.



Terme associé : **BIP-0047**

PBKDF2

CRYPTOGRAPHIE

`PBKDF2` est le sigle de *Password-Based Key Derivation Function 2*. C'est une méthode pour créer des clés cryptographiques à partir d'un mot de passe en utilisant une fonction de dérivation. Elle prend en entrée un mot de passe, un sel cryptographique, et applique de manière itérative une fonction prédéterminée (souvent une fonction de hachage comme `SHA256` ou un `HMAC`) sur ces données. Ce processus est répété de nombreuses fois afin de générer une clé cryptographique.

Dans le contexte de Bitcoin, `PBKDF2` est utilisée en conjonction avec la fonction `HMAC-SHA512` pour créer la graine d'un portefeuille déterministe et hiérarchique (seed) à partir d'une phrase de récupération de 12 ou de 24 mots. Le sel cryptographique utilisé dans ce cas est la chaîne `"mnemonic"` concaténée avec l'éventuelle passphrase BIP-0039, et le nombre d'itérations est de `2048`.

Termes associés : **HMAC-SHA512** **PHRASE DE RÉCUPÉRATION**

PEACH

ORGANISATION

Application mobile d'échange de bitcoins en pair à pair (P2P), fondée en 2022 en Suisse. Peach Bitcoin, souvent simplement appelée « Peach », permet aux utilisateurs d'acheter et de vendre des bitcoins directement entre eux sans passer par une plateforme d'échange centralisée.

L'application fonctionne avec un système d'*escrow* pour sécuriser les transactions et prend en charge de nombreuses méthodes de paiement en Europe, comme les virements SEPA, Revolut, PayPal ou les paiements en espèces lors de rencontres.

Termes associés : **HODL HODL** **LNP2PBOT**

PEDERSEN COMMITMENT

CRYPTOGRAPHIE

Un Pedersen commitment est un type d'engagement cryptographique présentant la propriété d'être homomorphique vis-à-vis de l'opération d'addition. Cela signifie qu'il est possible de valider la somme de deux engagements sans dévoiler les valeurs individuelles.

Formellement, si :

$$C_1 = commit(m_1, r_1) \quad C_2 = commit(m_2, r_2)$$

alors :

$$C_3 = C_1 \cdot C_2 = commit(m_1 + m_2, r_1 + r_2)$$

Cette propriété devient utile, par exemple, pour dissimuler les montants de *tokens* échangés dans des systèmes de cryptomonnaies, comme par exemple RGB, tout en pouvant vérifier les totaux.

Terme associé : **RGB**

PEELING CHAIN

CONFIDENTIALITÉ | FR : CHAÎNE D'ÉPLUCHAGE

Motif de transaction sur la blockchain dans lequel un montant important de bitcoins est progressivement distribué à travers une longue série de transactions successives. À chaque étape, une fraction est envoyée à un destinataire, tandis que le reste revient vers une nouvelle adresse de change. Le solde résiduel diminue ainsi de transaction en transaction : chaque montant est « pelé » (*peeled*) du total restant. La chaîne peut s'étendre sur des centaines, voire des milliers de transactions.

Ce motif est caractéristique des retraits automatisés depuis des plateformes d'échange, des pools de minage ou des services de paiement, mais il est également utilisé pour le blanchiment de fonds, car les petits montants individuels attirent moins l'attention des systèmes de conformité.

En analyse de chaîne, la *peeling chain* constitue un motif facile à identifier, car chaque transaction ne comporte généralement qu'une seule entrée et deux sorties aux montants très asymétriques. Son identification permet de suivre le fil des paiements et de reconstituer les flux de fonds à partir d'une source unique.

Termes associés : **ANALYSE DE CHAINE** | **HEURISTIQUE D'ANALYSE**

PEER DISCOVERY

RÉSEAU | FR : DÉCOUVERTE DES PAIRS

Processus par lequel les nœuds du réseau Bitcoin se connectent à d'autres nœuds pour obtenir des informations. Lorsqu'un nœud Bitcoin est lancé pour la première fois, il ne possède aucune information sur les autres nœuds du réseau. Pourtant, il doit établir des connexions pour se synchroniser sur la blockchain avec le plus de travail accumulé. Plusieurs mécanismes sont utilisés pour découvrir ces pairs, par ordre de priorité :

- Le nœud commence par consulter son fichier local `peers.dat`, qui stocke des informations sur les nœuds avec lesquels il a précédemment interagi. Si le nœud est nouveau, ce fichier sera vide, et le processus passera à l'étape suivante ;
- En l'absence d'informations dans le fichier `peers.dat` (ce qui est normal pour un nœud nouvellement lancé), le nœud effectue des requêtes DNS auprès des DNS seeds. Ces serveurs fournissent une liste d'adresses IP de nœuds a priori actifs pour établir des connexions. Les adresses des DNS seeds sont codées en dur dans le code de Bitcoin Core. Cette étape est généralement suffisante pour compléter la découverte des pairs ;
- Si les DNS seeds ne répondent pas dans les 60 secondes, le nœud peut alors se tourner vers les seed nodes. Ce sont des nœuds Bitcoin publics répertoriés dans une liste statique de près d'un millier d'entrées intégrée directement dans le code source de Bitcoin Core. Le nouveau nœud utilisera cette liste pour établir une première connexion au réseau et obtenir des adresses IP d'autres nœuds ;
- Dans le cas très peu probable où toutes les méthodes précédentes échouent, l'opérateur du nœud a toujours la possibilité d'ajouter manuellement des adresses IP de nœuds pour établir une première connexion.

Termes associés : **DNS SEEDS** | **SEED NODES**

PEERS.DAT

RÉSEAU

Fichier dans lequel Bitcoin Core sérialise les tables du module `addrman` : la table « new » (adresses découvertes mais pas encore testées) et la table « tried » (adresses auxquelles le nœud s'est déjà connecté avec succès). Il contient les adresses IP des autres nœuds, les numéros de ports, les horodatages de dernière connexion et la clé de bucketing (`nKey`) qui détermine le placement des adresses dans les seaux.

Lors du premier lancement, le fichier n'existe pas et les tables sont vides : le nœud interroge les DNS seeds pour obtenir une liste initiale de pairs. Au fil des connexions, les adresses sont insérées dans les tables et `peers.dat` est périodiquement mis à jour. Il est également possible d'ajouter manuellement des adresses via l'option `-addnode` ou la commande RPC `addnode`.

Ce fichier a remplacé `addr.dat` à partir de Bitcoin Core 0.7.0 (2012), lorsque Pieter Wuille a introduit `addrman` avec son système de bucketing déterministe.

Termes associés : **ADDR.DAT** | **ADDRMAN** | **PEER DISCOVERY**

PEG-IN

SIDECHAIN	FR : ANCRAGE ENTRANT

Mécanisme par lequel la valeur de bitcoins est transférée depuis la blockchain principale vers une sidechain. Le processus consiste à verrouiller des BTC sur la chaîne principale en les envoyant à une adresse spéciale gérée par un tiers de confiance ou un groupe, ou bien sur un script qui les rend temporairement inutilisables, puis à émettre une quantité équivalente de jetons sur la sidechain. Ce verrouillage garantit que les bitcoins ne peuvent pas être dépensés simultanément sur les deux chaînes. Une fois le peg-in effectué, l'utilisateur peut exploiter les fonctionnalités propres à la sidechain. Le peg-in constitue la première étape de l'ancrage bilatéral (*two-way peg*) qui relie la chaîne principale à la sidechain.

En tant que particulier, il n'est généralement pas nécessaire de recourir à un peg-in, puisqu'il existe des systèmes d'*atomic swap* permettant d'échanger des bitcoins *onchain* contre des bitcoins sur Liquid (ou sur une autre sidechain le cas échéant).

Termes associés : **PEG-OUT** **SIDECHAIN**

PEG-OUT

SIDECHAIN	FR : ANCRAGE SORTANT

Mécanisme inverse du peg-in, par lequel des jetons qui circulent sur une sidechain sont détruits afin de libérer les bitcoins correspondants sur la blockchain principale. L'utilisateur initie le processus en envoyant ses jetons vers une adresse de destruction sur la sidechain, ce qui déclenche le déverrouillage des BTC précédemment verrouillés sur la chaîne principale. Le peg-out constitue la seconde étape de l'ancrage bilatéral (*two-way peg*) et permet de revenir à tout moment sur la blockchain Bitcoin. Selon les implémentations, ce processus peut être géré par une fédération de signataires, comme c'est le cas sur Liquid, ou bien de manière automatisée via des preuves cryptographiques.

Termes associés : **PEG-IN** **SIDECHAIN**

PERCOLATION

</> INFORMATIQUE	EN : PERCOLATION

Fait référence à un modèle qui permet de comprendre la diffusion des informations (transactions et blocs) dans le réseau de nœuds Bitcoin. La théorie de la percolation est initialement un modèle mathématique et physique qui étudie le mouvement et la filtration de fluides à travers des matériaux poreux. Elle analyse comment, au-delà d'un certain seuil, un réseau connecté permet au fluide de s'écouler de manière continue à travers le matériau. On peut l'appliquer à des réseaux informatiques afin de voir comment les informations se diffusent en considérant les nœuds comme des sites pouvant être soit actifs, soit inactifs. Dans Bitcoin, les nœuds jouent ainsi le rôle des pores dans la théorie de la percolation. Chaque nœud actif reçoit et transmet l'information à d'autres nœuds qui vont soit continuer la transmission, soit la bloquer. La diffusion de certains types de transactions peut être analysée en termes de seuils de percolation, où un certain pourcentage de nœuds actifs est nécessaire pour atteindre un

mineur qui l'inclura dans un bloc. Cette théorie permet d'avoir un cadre pour évaluer comment les changements dans le réseau, comme la modification des règles de standardisation par certains nœuds, affectent le mécanisme de propagation en cascade des transactions pour atteindre un mineur.

Termes associés : **NOEUD** | **RÈGLES DE STANDARDISATION**

PÉRIMÉ - BLOC

PROTOCOLE | EN : STALE - BLOCK

Fait référence à un bloc sans enfant : un bloc valide, mais exclu de la chaîne principale de Bitcoin. Il se produit lorsque deux mineurs trouvent un bloc valide sur une même hauteur de chaîne durant un court laps de temps et le diffusent sur le réseau. Les nœuds finissent par choisir un seul bloc à inclure dans la chaîne, selon le principe de la chaîne avec le plus de travail accumulé, rendant l'autre « orphelin », « obsolète » ou « périmé ».

Termes associés : **MINEUR** | **ORPHELIN**

PÉRIODE DE MATURITÉ

PROTOCOLE | EN : MATURITY PERIOD

Délai nécessaire avant qu'une récompense de bloc ne soit dépensable par le mineur qui l'a reçue. Cette période est fixée à 100 blocs suivant le bloc miné, soit 101 confirmations pour la transaction coinbase. Pendant ce laps de temps, les bitcoins nouvellement créés dans la récompense de bloc ne sont pas dépensables. Cette règle a pour but d'éviter les complications liées à l'utilisation de bitcoins issus d'une chaîne qui pourrait être ultérieurement rendue obsolète. En effet, il arrive que des blocs valides soient finalement invalidés si un autre bloc, à la même hauteur, est intégré dans une chaîne bénéficiant de plus de travail. Ce phénomène, appelé réorganisation, aboutit à la création d'un « bloc orphelin » ou « bloc obsolète », privant ainsi le mineur des bitcoins contenus dans la coinbase du bloc abandonné. Si les bitcoins nouvellement créés étaient immédiatement dépensables, toute transaction les impliquant pourrait être annulée a posteriori, causant des pertes pour les détenteurs de ces bitcoins. Un tel scénario pourrait entraîner des annulations en série de transactions pourtant valides, affectant ainsi tous les utilisateurs impliqués dans cette chaîne de transactions. La période de maturité est donc un mécanisme de prévention contre ce risque. En imposant un délai de 100 blocs avant que les bitcoins nouvellement émis puissent être utilisés, on évite que des pièces issues de blocs finalement invalidés ne perturbent le système en circulant et en affectant d'autres transactions. La probabilité de voir survenir une réorganisation de 101 blocs est si faible qu'elle est considérée comme nulle.

Termes associés : **COINBASE** | **RÉCOMPENSE DE BLOC**

PHOENIX

LIGHTNING NETWORK

Logiciel de portefeuille mobile *self-custodial* conçu pour simplifier et rendre plus accessible les transactions sur le protocole Lightning. Il permet aux utilisateurs de gérer leurs fonds directement depuis leurs appareils mobiles sans nécessiter la connexion à un nœud Lightning séparé. Phoenix fonctionne en fait comme un véritable nœud Lightning autonome sur le téléphone (implémentation *lightning-kmp*). Il prend en charge les transactions Bitcoin et Lightning, et offre des fonctionnalités pour faciliter la gestion du nœud, telles que la gestion automatique des canaux avec le nœud d'Acinq. Contrairement aux autres applications de portefeuille Lightning, qui sont pour la plupart custodiales, Phoenix offre un compromis intéressant en combinant l'utilisation d'un nœud Lightning avec la commodité d'une application pour smartphone. C'est une des meilleures solutions simples à prendre en main pour un débutant qui souhaite utiliser Lightning tout en conservant la pleine possession de ses bitcoins (*self-custody*). Phoenix est un projet développé et maintenu par l'entreprise française Acinq.

Termes associés : **ECLAIR** | **PHOENIXD**

PHOENIXD

LIGHTNING NETWORK

Implémentation spécialisée d'un nœud Lightning (éclair), conçue pour envoyer et recevoir des paiements depuis un serveur. Phoenixd utilise le même logiciel que le portefeuille/nœud mobile Phoenix, mais à la différence de ce dernier, qui est exclusivement destiné aux smartphones, phoenixd est capable de fonctionner sur un serveur sous forme de *daemon*. Au lieu de posséder une interface graphique utilisateur (GUI), il est équipé d'une interface API HTTP et prend en charge la gestion automatique des canaux et de la liquidité. Phoenixd est un projet développé et maintenu par l'entreprise française ACINQ.

Termes associés : **ECLAIR** | **PHOENIX**

PHRASE DE RÉCUPÉRATION

PORTEFEUILLE | EN : MNEMONIC PHRASE

Une phrase de récupération, également parfois nommée mnémonique, *seed phrase*, ou phrase secrète, est une séquence composée habituellement de 12 ou de 24 mots, qui est générée de manière pseudo-aléatoire à partir d'une source d'entropie. La séquence pseudo-aléatoire est toujours complétée d'une somme de contrôle (*checksum*). La phrase mnémonique, conjointement avec une passphrase optionnelle, est utilisée pour dériver de façon déterministe l'intégralité des clés associées à un portefeuille HD (déterministe et hiérarchique). Cela signifie qu'à partir de cette phrase, il est possible de générer et de recréer déterministiquement l'ensemble des clés privées et publiques du portefeuille Bitcoin, et par conséquent d'accéder aux fonds qui y sont associés. La raison d'être de la phrase de récupération est de fournir un moyen de sauvegarde et de récupération des bitcoins qui est à la fois sécurisé et facile à utiliser.

Il est important de conserver cette phrase en lieu sûr et de manière sécurisée, car toute personne en possession de la mnémonique aurait accès aux

fonds du portefeuille correspondant. Si elle est utilisée dans le cadre d'un portefeuille classique, et sans passphrase optionnelle, elle constitue souvent un *SPOF* (point de défaillance unique). La phrase de récupération est donc un encodage de la séquence pseudo-aléatoire et de la *checksum* dans des mots du quotidien afin de faciliter sa notation et sa lisibilité par l'Homme. Elle est construite en fonction du standard BIP-0039, qui définit et ordonne une liste de 2048 mots utilisés pour cet encodage.

La liste de 2048 mots du BIP-0039 est disponible dans plusieurs langues, toutefois, il est conseillé d'utiliser uniquement la version anglaise, car c'est la version la plus largement prise en charge par les logiciels de portefeuille.

Termes associés : **BIP-0039** | **PASSPHRASE - BIP-0039**

PILE

SCRIPT | EN : STACK

Dans le contexte du langage script utilisé pour apposer des conditions de dépense sur des UTXOs Bitcoin, la pile est une structure de données de type « LIFO » (*Last In, First Out*) qui sert à stocker des éléments temporaires pendant l'exécution d'un script. Chaque opération dans le script manipule ces piles, où les éléments peuvent être ajoutés (*push*) ou retirés (*pop*). Les scripts utilisent les piles pour évaluer les expressions, stocker des variables temporaires, et gérer les conditions.

Dans l'exécution d'un script Bitcoin, 2 piles peuvent être utilisées : la pile principale et la pile alt (alternative). La pile principale est utilisée pour la majorité des opérations d'un script. C'est sur cette pile que les opérations de script ajoutent, retirent ou manipulent des données. La pile alternative, quant à elle, sert à stocker temporairement des données pendant l'exécution du script. Certains opcodes spécifiques, comme `OP_TOALTSTACK` et `OP_FROMALTSTACK`, permettent de transférer des éléments de la pile principale vers la pile alternative et inversement.

Par exemple, lors de la validation d'une transaction, les signatures et les clés publiques sont poussées sur la pile principale et traitées par des opcodes successifs pour vérifier que les signatures correspondent aux clés et aux données de la transaction.

Termes associés : **OPCODES** | **SCRIPT**

PILULE ORANGE

HISTOIRE | EN : ORANGE PILL

Inspirée du film *Matrix*, cette pilule imaginaire éveille à la conscience de Bitcoin.

S'utilise également comme verbe. Orange-piller quelqu'un = l'enrôler dans Bitcoin.

Termes associés : **BITCOINTALK** | **HODL**

PIZZA DAY

HISTOIRE

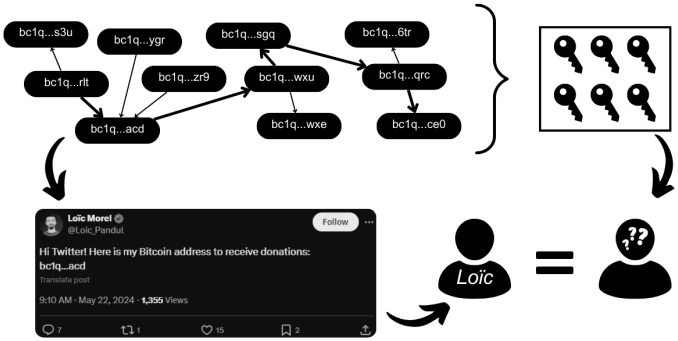
Évènement célébré chaque 22 mai par la communauté Bitcoin qui commémore la première transaction de bitcoins contre un bien physique. En 2010, Laszlo Hanyecz, un développeur Bitcoin, a proposé sur le forum BitcoinTalk d'acheter deux grandes pizzas pour 10 000 BTC, alors équivalents à environ 40 dollars. Le 22 mai, il a confirmé que l'offre avait été acceptée par un étudiant californien de 19 ans, Jeremy Sturdivant, connu sous le pseudonyme de Jercos. Celui-ci a commandé et fait livrer les pizzas de chez Papa John's à Laszlo en Floride. Ce jour marque un moment important pour Bitcoin puisqu'il démontre son potentiel en tant que monnaie d'échange. Chaque année, la communauté célèbre cet événement en consommant des pizzas payées en bitcoins.

Termes associés : **BITCOINTALK** | **NAKAMOTO SATOSHI**

POINT D'ENTRÉE

CONFIDENTIALITÉ | EN : ENTRY POINT

Information permettant de lier une activité *on-chain* (une adresse, une transaction, un *cluster*...) à une forme d'identité appartenant à un utilisateur ou à une entité. Par exemple, si vous publiez votre adresse de réception sur Twitter sous votre nom, un analyste pourrait la retrouver et l'associer à votre identité. Dans ce cas, le tweet constituerait un point d'entrée pour une analyse de chaîne. Pour identifier un point d'entrée, les analystes peuvent utiliser l'OSINT, mais la méthode la plus répandue pour associer une activité *on-chain* à une identité reste le KYC.



Termes associés : **ANALYSE DE CHAINE** | **KYC - KNOW YOUR CUSTOMER**

POINT GÉNÉRATEUR

CRYPTOGRAPHIE | EN : GENERATOR POINT

Point fixe sur une courbe elliptique qui sert de base pour toutes les opérations de cryptographie à clé publique. Dans Bitcoin, le point générateur `G` est un paramètre standard de la courbe secp256k1 défini de manière déterministe. Tous les utilisateurs de Bitcoin partagent le même point générateur `G`. C'est un paramètre public et universel de la courbe secp256k1. Ses coordonnées sont :

```
x=0×79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F8
1798
y=0×483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10
D4B8
```

Le point générateur est à la base de la dérivation des clés publiques : une clé privée `k` (un nombre entier) est associée à une clé publique `K` par multiplication scalaire sur la courbe elliptique : `K = k * G`. Cette opération est facile à calculer dans un sens, mais pratiquement impossible à inverser (problème du logarithme discret sur courbes elliptiques).

Termes associés : **COURBE ELLIPTIQUE** | **SECP256K1**

POLAR

OUTIL

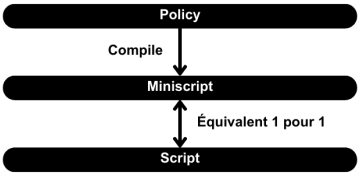
Outil de simulation locale du Lightning Network destiné au développement et aux tests. Développé par Jamal James, Polar permet de créer rapidement un réseau Lightning complet sur son ordinateur, avec plusieurs nœuds LND, CLN ou Eclair interconnectés, sans avoir besoin de vrais bitcoins ni de connexion au réseau Bitcoin. L'interface graphique permet de visualiser les canaux, d'envoyer des paiements de test et de simuler différents scénarios de routage, ce qui en fait un outil précieux pour les développeurs d'applications Lightning.

Termes associés : **LIGHTNING NETWORK** | **LND**

POLICY - MINISCRIPT

SCRIPT | FR : POLITIQUE - MINISCRIPT

Langage de haut niveau orienté utilisateur permettant de spécifier simplement des conditions sous lesquelles un UTXO peut être débloqué dans le cadre de Miniscript. La *policy* est une description abstraite des règles de dépense. Elle peut ensuite être compilée en miniscript, qui est équivalent un pour un avec des opérations du langage script natif de Bitcoin.



Le langage de *policies* est légèrement différent du langage miniscript. Par

exemple, imaginons un système de sécurisation avec en chemin primaire, la clé A, et en chemin de récupération, la clé B, mais sous un *timelock* d'un an (environ 52 560 blocs). En *policy*, cela donnerait :

```
or(pk(A),and(pk(B),older(52560)))
```

Une fois compilé en miniscript, cela donnerait :

```
andor(pk(B),older(52560),pk(A))
```

Et une fois converti en script natif, cela donnerait :

```
<B>
OP_CHECKSIG
OP_NOTIF
<A>
OP_CHECKSIG
OP_ELSE
<50cd00>
OP_CHECKSEQUENCEVERIFY
OP_ENDIF
```

Terme associé : **MINISCRIPT**

POLITIQUE DE MEMPOOL

RÉSEAU | EN : MEMPOOL POLICY

Sous-ensemble de la politique de relais des transactions qui regroupe l'ensemble des règles appliquées par un nœud Bitcoin pour déterminer quelles transactions non confirmées il accepte dans sa mempool. La politique de mempool englobe les règles de standardisation, mais s'étend au-delà en incluant des règles qui prennent en compte le contexte de la mempool dans son ensemble, et pas seulement le format des transactions individuelles.

Par exemple, les règles relatives au taux de frais minimum de relais (`minRelayTxFee`), à la taille maximale de la mempool (`maxmempool`), ou encore les règles de remplacement de transactions (RBF) relèvent de la politique de mempool. Ces règles sont configurées localement par chaque nœud et peuvent varier d'un nœud à l'autre.

La politique de mempool se distingue des règles de standardisation en ce que ces dernières ne concernent que le format d'une transaction prise isolément, tandis que la politique de mempool prend en compte des facteurs contextuels comme l'encombrement de la mempool, les relations entre transactions, ou les seuils de frais configurés par l'opérateur du nœud.

Terme associé : **RÈGLES DE STANDARDISATION**

POLITIQUE DE RELAIS

RÉSEAU | EN : RELAY POLICY

Ensemble de toutes les règles observées par le logiciel d'un nœud Bitcoin lors de ses interactions avec le réseau pair-à-pair. La politique de relais est le terme

le plus englobant pour désigner les règles locales d'un nœud, par opposition aux règles de consensus qui sont universelles.

La politique de relais se décompose en sous-ensembles de plus en plus spécifiques : *La* politique de relais des transactions* concerne spécifiquement les règles liées au traitement des transactions sur le réseau ; *La* politique de mempool* regroupe les règles d'acceptation des transactions dans la mempool du nœud ; *Les* règles de standardisation* dictent le format des transactions individuelles acceptées.

Contrairement aux règles de consensus, la politique de relais est configurée localement par chaque nœud et peut varier de l'un à l'autre.

Termes associés : **POLITIQUE DE RELAIS DES TX** | **RÈGLES DE CONSENSUS**

POLITIQUE DE RELAIS DES TX

RÉSEAU | EN : TRANSACTION RELAY POLICY

Sous-ensemble de la politique de relais qui regroupe l'ensemble des règles appliquées par un nœud Bitcoin spécifiquement pour le traitement et la retransmission des transactions sur le réseau pair-à-pair. La politique de relais des transactions englobe notamment la politique de mempool et, par extension, les règles de standardisation.

Ces règles déterminent si une transaction reçue d'un pair sera acceptée, stockée dans la mempool, puis retransmise aux autres nœuds du réseau. Elles vont au-delà du simple format des transactions individuelles (couvert par les règles de standardisation) et incluent également des considérations liées à la gestion du réseau.

Comme toutes les règles de politique, la politique de relais des transactions est configurée localement par chaque nœud et peut varier de l'un à l'autre. Elle se distingue de la politique de relais au sens large, qui couvre également des aspects non liés aux transactions, comme le traitement des blocs ou la gestion des connexions avec les pairs.

Termes associés : **POLITIQUE DE MEMPOOL** | **POLITIQUE DE RELAIS**

POOL DE COINJOIN

CONFIDENTIALITÉ

Dans le contexte des coinjoins, une pool est un groupe d'utilisateurs qui s'accorde à mélanger leurs UTXOs de manière homogène pour casser leurs historiques. Une pool de coinjoin impose un montant fixe pour les UTXOs participants, ce qui garantit que les inputs et les outputs des transactions coinjoins sont identiques, et empêche ainsi de faire un lien entre eux. Au plus il y a de participants dans une pool, au plus les pièces peuvent y gagner de grands anonsets en réalisant plusieurs cycles de remixage.

Terme associé : **COINJOIN**

POOL HOPPING

↗ MINAGE | FR : SAUT DE COOPÉRATIVES

Désigne la pratique de certains mineurs consistant à changer fréquemment de pool de minage pour maximiser leurs gains. Ces mineurs passent d'une pool à une autre en fonction des variations de la rentabilité. Cette stratégie exploite les différences dans les méthodes de calcul des récompenses des pools. Le *pool hopping* peut déséquilibrer la distribution des récompenses au sein des pools et est généralement considéré comme une pratique déloyale dans la communauté.

Termes associés : **MINING POOL** | **SHARES**

POOL - LIGHTNING

ϟ LIGHTNING NETWORK

Service développé par Lightning Labs. C'est une sorte de marché pour les liquidités dans les canaux Lightning. Pool connecte les utilisateurs ayant besoin d'accès à la liquidité sur LN avec ceux qui ont des capitaux à déployer sur le réseau. Les participants peuvent gagner des satoshis en ouvrant de nouveaux canaux pour ceux qui cherchent à recevoir des fonds sur Lightning pendant une période déterminée. Pool permet également de louer un canal pour accepter instantanément des paiements Bitcoin.

Termes associés : **CANAL DE PAIMENT** | **LIQUIDITÉS**

PORTAL

⊸ PORTEFEUILLE

Hardware wallet Bitcoin *open source* développé par TwentyTwo Devices, qui fonctionne entièrement via NFC, sans câble ni batterie. L'appareil s'alimente directement par le champ NFC du *smartphone*. Son *firmware* est écrit en Rust avec BDK (*Bitcoin Dev Kit*) et il est compatible avec des portefeuilles mobiles comme Nunchuk ou Bitcoin Keeper pour signer des transactions.

Termes associés : **BDK - BITCOIN DEV KIT** | **TWENTYTWO DEVICES**

PORTE DÉROBÉE

✸ ATTAQUE | EN : BACKDOOR

Une *backdoor* est un mécanisme secret qui permet de disposer d'un accès privilégié à un système informatique, un logiciel, une fonction, un algorithme ou des données, sans passer par les procédures d'authentification ou de sécurité habituelles. À la différence d'une faille de sécurité, les portes dérobées sont introduites intentionnellement dans le code source par des développeurs malveillants. Elles peuvent être utilisées pour espionner, manipuler ou voler des informations sensibles.

Le terme de « porte dérobée » est assez peu utilisé en français. On préfère généralement employer directement le terme anglais « backdoor ».

Termes associés : **FOSS** | **SURFACE D'ATTAQUE**

PORTEFEUILLE

PORTEFEUILLE | EN : WALLET

Outil logiciel spécialement conçu pour sécuriser et gérer les clés privées d'un utilisateur. Si le portefeuille est stocké et géré sur un dispositif logiciel lui-même installé sur une machine polyvalente, on parle alors de « portefeuille chaud ». En revanche, s'il est stocké dans un logiciel, lui-même installé sur un dispositif matériel dédié uniquement à cette tâche et non connecté à internet, on parle alors de « portefeuille froid ». Le portefeuille permet notamment d'utiliser les clés privées pour signer des transactions, et ainsi remplir les conditions permettant la dépense des bitcoins.

En français, beaucoup utilisent directement le terme anglais « wallet » pour évoquer un portefeuille.

Termes associés : **COLD WALLET** | **PORTEFEUILLE CHAUD**

PORTEFEUILLE CHAUD

PORTEFEUILLE | EN : HOT WALLET / SOFTWARE WALLET

Un portefeuille chaud est un dispositif logiciel dédié à la sécurisation et à la gestion des clés privées d'un portefeuille Bitcoin. On parle de portefeuille chaud lorsque la phrase de récupération d'un portefeuille Bitcoin est conservée sur un appareil informatique, via un logiciel, qui n'est pas dédié uniquement à une utilisation de Bitcoin et qui est connecté directement à internet. Par exemple, une application de portefeuille Bitcoin installée sur votre smartphone serait considérée comme un portefeuille chaud.

Termes associés : **COLD WALLET** | **PORTEFEUILLE**

POSTMIX

CONFIDENTIALITÉ

Compte qui regroupe les UTXOs qui ont participé à au moins un cycle de coinjoin dans le protocole Whirlpool. Le postmix représente les fonds mixés dont le lien avec leur origine a été rompu grâce au processus de coinjoin.

Les UTXOs dans le compte postmix bénéficient d'anonsets qui augmentent avec chaque cycle de coinjoin supplémentaire. Plus un UTXO participe à des remixages (qui sont gratuits dans Whirlpool), plus son ensemble d'anonymat prospectif et rétrospectif grandit, ce qui renforce sa confidentialité.

L'utilisation des UTXOs postmix requiert une attention particulière pour ne pas dégrader la confidentialité acquise. Fusionner des UTXOs postmix entre eux ou avec des UTXOs non mixés peut rétablir des liens traçables. Les dépenses depuis le postmix gagnent à utiliser des techniques préservant la confidentialité, comme Stonewall ou Stonewall x2.

Termes associés : **ANONSETS - ANONYMITY SETS** | **WHIRLPOOL**

POT - PAY ON TARGET

↗ MINAGE

Méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. POT est un système de rémunération variant selon la difficulté du travail envoyé à la pool plutôt que celle du travail fourni par la pool. Dans cette approche, la récompense pour chaque part soumise par un mineur est établie sur la difficulté de cette part spécifique. Cela signifie que les parts plus difficiles sont mieux récompensées que celles moins difficiles. C'est une méthode variante de PPS, mais en ajustant la récompense en fonction de la complexité réelle du travail accompli, POT implique beaucoup plus de variance pour le mineur.

Terme associé : **SHARES**

PPA

↗ MINAGE | FR : CONTRAT D'ACHAT D'ÉNERGIE

Sigle de « *Power Purchase Agreement* ». C'est un contrat à long terme entre une grosse ferme de minage de bitcoins et un fournisseur d'énergie, pour l'achat direct d'électricité à un tarif fixe. Ce type de contrat permet aux fermes de minage de sécuriser des coûts énergétiques stables et compétitifs sur plusieurs années, ce qui réduit l'impact des fluctuations du marché de l'électricité.

Termes associés : **FERME DE MINAGE** **MINAGE**

PPLNS - PAY PER LAST N SHARES

↗ MINAGE

Méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. PPLNS récompense les mineurs en fonction de leur contribution en *shares* sur une période donnée. Dans PPLNS, les paiements sont effectués seulement lorsque la pool trouve un bloc et sont établis sur le nombre de *shares* soumises par le mineur par rapport au total des *shares* collectées pendant la période observée. Cette méthode favorise les mineurs constants et actifs sur le long terme, car elle décourage le *pool hopping* (changement fréquent de pool). La rémunération varie avec la probabilité de trouver un bloc, ce qui peut entraîner une baisse de la constance dans les revenus du mineur (plus de variance que le paiement à la tâche).

Terme associé : **SHARES**

PPS - PAY PER SHARE

↗ MINAGE

Méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. PPS est un système où les mineurs sont rémunérés pour chaque share valide soumise, indépendamment du fait que la pool trouve ou non un bloc. Ils sont donc rémunérés sur la valeur attendue. C'est une méthode de rémunération à la tâche.

Chaque share soumise est considérée comme une contribution au processus de minage et a une valeur fixe prédéterminée. Cette méthode offre une rémunération stable et prévisible pour les mineurs, car elle élimine la variance liée à la probabilité de trouver un bloc. Toutefois, elle est plus risquée pour les opérateurs de pool, car ils doivent payer les mineurs même lorsque aucun bloc n'est trouvé, absorbant ainsi le risque de variance. Contrairement à la méthode FPPS, PPS n'inclut pas les frais de transaction dans le calcul de la rémunération.

Terme associé : **SHARES**

PPS+

↗ MINAGE

Sigle de « *Pay Per Share Plus* ». Méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage, qui combine les approches PPS et PPLNS. Pour la partie liée à la subvention de bloc, les mineurs reçoivent une rémunération fixe pour chaque *share* valide soumise, calculée selon la méthode PPS classique, indépendamment du fait que la pool ait trouvé un bloc ou non. Pour la partie liée aux frais de transaction, la rémunération est distribuée selon la méthode PPLNS, uniquement lorsqu'un bloc est effectivement trouvé par la pool.

PPS+ offre ainsi aux mineurs une base de revenus stable grâce à la composante PPS, tout en leur permettant de bénéficier des frais de transaction lorsque la pool trouve un bloc. Cette méthode est moins risquée pour l'opérateur de la pool que le FPPS, car il n'avance pas les frais de transaction.

Termes associés : **PPLNS - PAY PER LAST N SHARES** | **PPS - PAY PER SHARE**

PRAXÉOLOGIE

$ ÉCONOMIE ET RÉGULATION | EN : PRAXEOLOGY

Science générale de l'action humaine, développée par Ludwig von Mises dans le cadre de l'École autrichienne d'économie. La praxéologie étudie le comportement intentionnel et orienté vers une fin. Elle repose sur un axiome fondamental : l'individu agit délibérément pour passer d'un état qu'il juge moins satisfaisant à un état qu'il perçoit comme meilleur. Cet axiome est considéré comme irréfutable, car toute tentative de le contredire constitue elle-même une action, ce qui le valide. L'économie, en tant que catallactique, est une branche de la praxéologie.

La méthode praxéologique est axiomatique et déductive : elle part de propositions considérées comme vraies a priori pour en déduire logiquement des lois

économiques, sans recourir à l'expérimentation empirique ni à l'économétrie. Cette approche se distingue radicalement du positivisme des autres courants économiques. Elle implique notamment que les prix, la monnaie et les cycles économiques sont analysés comme des phénomènes émergents de l'action individuelle, et non comme des agrégats statistiques.

Dans l'écosystème Bitcoin, la praxéologie est souvent mobilisée pour analyser le bitcoin comme une monnaie émergente, dont l'adoption et la valeur découlent des choix volontaires d'individus agissant dans leur intérêt propre, sans planification centralisée.

Termes associés : **PRÉFÉRENCE TEMPORELLE** | **ÉCOLE AUTRICHIENNE**

PRÉFÉRENCE TEMPORELLE

Ⓢ ÉCONOMIE ET RÉGULATION | EN : TIME PREFERENCE

Concept de l'économie autrichienne qui décrit la tendance des individus à préférer la satisfaction immédiate à la satisfaction future. Un individu avec une préférence temporelle élevée privilégie la consommation présente ; un individu avec une préférence temporelle basse accepte de reporter sa consommation pour obtenir un bénéfice supérieur dans le futur. Ce concept est central dans la théorie autrichienne de l'intérêt et du capital.

La préférence temporelle est directement liée à la qualité de la monnaie utilisée. Une monnaie qui perd sa valeur dans le temps (par l'inflation) incite à une préférence temporelle élevée : il est rationnel de dépenser rapidement plutôt que d'épargner. À l'inverse, une monnaie dure, dont la valeur se maintient ou s'apprécie, encourage l'épargne, la planification à long terme et l'investissement productif.

Bitcoin, par sa rareté absolue et sa politique monétaire désinflationniste, est souvent présenté comme un outil qui favorise une préférence temporelle basse. Il inciterait ses détenteurs à adopter une vision de long terme, tant dans leurs choix financiers que dans leur comportement de consommation.

Terme associé : **ÉCOLE AUTRICHIENNE**

PRÉFIXES BINAIRES

</> INFORMATIQUE | EN : BINARY PREFIXES

Unités utilisées en informatique pour quantifier les multiples de tailles de données établis sur des puissances de 2. Contrairement aux préfixes du système métrique qui utilisent une base de 10, les préfixes binaires, tels que kibi (Ki), mebi (Mi), gibi (Gi), et tebi (Ti), multiplient par des puissances de 2 (2^{10}, 2^{20}, 2^{30}, 2^{40} respectivement). Ces préfixes sont hérités des premières manières de mesurer la taille d'informations sur des ordinateurs. On les retrouve parfois dans Bitcoin, comme par exemple pour désigner la limite de taille des fichiers `BLOCKS/BLK*.DAT` qui permettent de stocker les données brutes de la blockchain dans le logiciel Bitcoin Core. Ces derniers disposent ainsi d'une capacité maximale de 128 mébioctets (Mio), ce qui équivaut à un peu plus de 134 mégaoctets (Mo).

Termes associés : **BIT** | **OCTET**

PREIMAGE

CRYPTOGRAPHIE | FR : PRÉIMAGE

En mathématiques, la préimage d'un élément y par une fonction f est l'ensemble des x tels que $f(x) = y$. C'est l'opération d'inverser la fonction : remonter de la sortie vers l'entrée.

En cryptographie, ce concept est à la base de la sécurité des fonctions de hachage : la résistance à la préimage impose qu'étant donné un hash h, il soit computationnellement infaisable de trouver m tel que $H(m) = h$. La fonction H agit comme une fonction à sens unique : facile à calculer, quasi impossible à inverser.

Sur Bitcoin, ce terme revient dès que l'on parle de fonction de hachage, mais il prend un sens plus spécifique sur Lightning, où la préimage désigne la valeur secrète aléatoire dont le hash sert de verrou pour les paiements. Lorsqu'un destinataire émet une invoice, il génère une *preimage* de 32 octets, en calcule le hash SHA-256 (le *payment hash*) et l'inclut dans l'invoice. Les HTLC établis le long de la route du paiement sont verrouillés par ce hash. Le paiement n'est finalisé que lorsque le destinataire révèle la *preimage* au nœud qui le précède sur la route. Celui-ci peut alors débloquer le HTLC correspondant, et la *preimage* se propage en sens inverse, permettant à chaque nœud intermédiaire de réclamer ses fonds. La révélation de la *preimage* constitue ainsi la preuve cryptographique que le paiement a bien été reçu.

Termes associés : **HTLC** | **PAYMENT HASH**

PREMIUM

COMMUNAUTÉ | FR : MAJORATION

Montant supplémentaire payé au-dessus du prix standard ou nominal d'un actif. Dans le contexte de Bitcoin, un premium peut être observé lors des achats, notamment sur les plateformes d'échange, qui peuvent parfois utiliser cette technique pour faire leurs marges sur le service de courtage.

On peut également retrouver cette notion de premium lors des achats de BTC en peer-to-peer. En effet, l'achat de bitcoins non-KYC va souvent avec un premium par rapport au prix standard du marché qui peut varier de 1 % jusqu'à parfois plus de 10 %. Plusieurs raisons expliquent cette différence de prix. D'abord, il s'agit d'une pratique courante chez les vendeurs P2P qui s'est installée au fil du temps. Ensuite, les vendeurs ont des frais associés à la transaction pour envoyer les fonds à l'acheteur. Il y a aussi un risque de vol accru lors de ventes en P2P par rapport aux transactions sur des plateformes régulées, ce qui justifie une compensation pour le risque pris. Enfin, le surcoût peut être lié à la demande et à la qualité de l'échange en termes de confidentialité. En tant qu'acheteur, le gain de confidentialité a un prix qui se reflète dans la majoration appliquée par le vendeur. Certains bitcoiners pensent également que le prix majoré du BTC acheté en P2P reflète son véritable cours, et avancent l'argument que les prix plus bas sur les plateformes régulées sont le résultat d'un compromis sur la confidentialité de vos données personnelles.

Termes associés : **KYC - KNOW YOUR CUSTOMER** | **SELF-CUSTODY**

PREMIX

CONFIDENTIALITÉ

Compte intermédiaire dans le protocole Whirlpool qui contient les UTXOs issus de la Tx0 en attente de leur premier cycle de coinjoin. Le premix est un état transitoire : les UTXOs y séjournent jusqu'à ce qu'ils soient sélectionnés pour participer à une transaction coinjoin.

Chaque UTXO dans le compte premix possède exactement le montant nominal de la pool choisie (par exemple 0,01 BTC pour la pool 0,01), en plus des frais de minage nécessaires pour le premier cycle de coinjoin. Ces UTXOs sont considérés comme non mixés, car ils sont encore directement traçables à la Tx0 qui les a créés. Ils n'ont pas encore bénéficié de la rupture de lien qu'offre le coinjoin.

Une fois qu'un UTXO premix est sélectionné pour un cycle de coinjoin et que la transaction est confirmée, il passe dans le compte postmix.

Termes associés : **TX0** **WHIRLPOOL**

PREUVE DE RÉSERVES

ÉCONOMIE ET RÉGULATION | EN : PROOF OF RESERVES

Mécanisme utilisé par les plateformes d'échanges de bitcoins pour démontrer qu'elles détiennent bien les actifs de leurs clients. Le but recherché est de prouver la solvabilité de l'entreprise en vérifiant que ses bitcoins sont équivalents ou supérieurs aux soldes des utilisateurs qu'ils sont censés détenir.

Les plateformes utilisent souvent des arbres de Merkle qui incluent toutes les balances de leurs utilisateurs. Chaque utilisateur peut alors vérifier que son solde fait partie de l'ensemble total, sans révéler d'informations sur les autres comptes. En parallèle, la plateforme signe publiquement les adresses qui sécurisent les bitcoins pour prouver qu'elles en ont le contrôle.

Cependant, ces méthodes présentent des limites importantes, car la preuve de possession des fonds ne prouve pas pour autant la solvabilité de l'entreprise. L'exchange pourrait, par exemple, lever des dettes en secret en contractant des prêts lombards garantis par les mêmes bitcoins utilisés dans la preuve de réserve. Il pourrait également afficher un solde en bitcoins empruntés, tout en restant insolvable. Le problème de la preuve de réserve est qu'elle se concentre uniquement sur la vérification des actifs de la plateforme, sans prendre en compte les passifs.

Traditionnellement sur Bitcoin, pour célébrer l'anniversaire de la publication du bloc de Genèse, chaque 3 janvier représente le « *Proof-of-Keys Day* ». Les détenteurs de bitcoins sont invités à tous retirer leurs fonds des exchanges pour les stocker dans des portefeuilles en self-custody, afin de tester la solvabilité de ces plateformes.

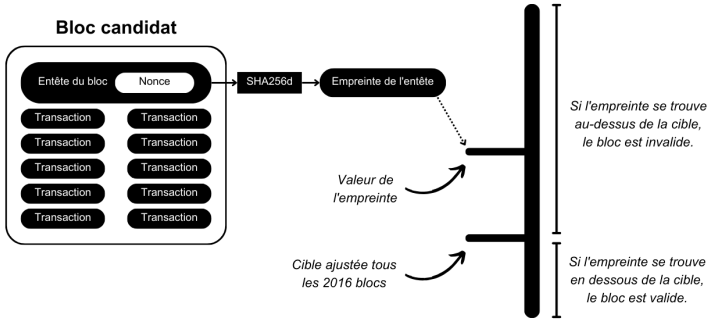
Mais finalement, si vous souhaitez réellement vous protéger face au risque de faillite des plateformes d'échange, la seule solution est de conserver vos bitcoins en self-custody.

Termes associés : **ARBRE DE MERKLE** **PROOF-OF-KEY DAY**

PREUVE DE TRAVAIL

PROTOCOLE | EN : POW - PROOF OF WORK

Mécanisme de protection face aux attaques Sybil, qui se caractérisent par la multiplication de fausses identités, dans le but de prendre un avantage illégitime. Ainsi, la preuve de travail permet d'établir un coût marginal non négligeable à la multiplication des votes sur Bitcoin. La preuve de travail est à la base du mécanisme de consensus de Nakamoto, qui est le principe utilisé pour établir un accord sur une version unique du registre distribué entre les différents nœuds du réseau. Concrètement, la preuve de travail est la recherche d'une valeur qui, une fois passée dans une fonction mathématique aléatoire, donne un résultat inférieur ou égal à un nombre cible. Cette cible de la preuve de travail est ajustée tous les 2016 blocs par les nœuds. C'est ce que l'on appelle l'ajustement de la difficulté. On abaisse le nombre cible pour augmenter la difficulté de minage, ou on l'augmente pour baisser la difficulté, en fonction de l'évolution de la puissance de calcul déployée par les mineurs durant la période précédente.



Ce travail effectué par les mineurs est récompensé à chaque bloc valide trouvé. Le mineur gagnant empoche une récompense pécuniaire, composée de la subvention de bloc (création de nouveaux bitcoins ex-nihilo), et des frais de transaction. Aujourd'hui, la difficulté de la preuve de travail sur Bitcoin est telle que le minage nécessite une grande puissance de calcul pour parvenir à gagner des blocs. En conséquence, il faut souvent disposer de puces électroniques spécialisées dans l'exécution de `SHA256d`, que l'on appelle des ASICs, et participer à des pools de minage.

Termes associés : **AJUSTEMENT DE LA DIFFICULTE** | **MINAGE**

PRIME DE PROXIMITÉ

PROTOCOLE | EN : PROXIMITY PREMIUM

Dans le cadre du minage de Bitcoin, fait référence à l'avantage financier que les mineurs obtiennent en réduisant la latence des communications sur le réseau. Les mineurs situés plus près les uns des autres reçoivent les nouveaux blocs valides plus rapidement, ce qui leur permet de cesser de gaspiller des ressources sur un bloc candidat qui se trouve sur une branche avec moins de travail accumulé. Cette réduction de la latence améliore les rendements des mineurs, ce qui les incite naturellement à se regrouper géographiquement pour maximiser leurs revenus.

La prime de proximité est un défaut inhérent au fonctionnement de Bitcoin, qui découle de deux mécanismes : la linéarité de l'ordre des blocs et le fait que la récompense de bloc est exclusivement attribuée au mineur qui trouve le bloc. Ce phénomène incite donc au regroupement des activités de minage.

Termes associés : **LATENCE** | **MINAGE**

PRIVATE CHANNEL

LIGHTNING NETWORK | FR : CANAL PRIVÉ

Canal Lightning qui n'est pas annoncé au réseau et donc absent du graphe des nœuds publics. Contrairement à un canal public classique, un canal privé ne diffuse pas de message d'annonce (*channel announcement*) aux autres pairs. Il est invisible pour les nœuds tiers et ne peut pas servir au routage de paiements pour d'autres utilisateurs.

Les canaux privés constituent le type de canal recommandé pour les nœuds qui ne font pas de routage, comme les nœuds embarqués sur mobiles. Pour recevoir un paiement via un canal privé, l'existence du canal et ses paramètres sont révélés au payeur à travers des *routing hints* intégrés dans la facture Lightning.

Le terme « privé » est ici un peu trompeur : il signifie simplement « non annoncé ». Des heuristiques on-chain, comme l'analyse des transactions de financement ou les fermetures forcées, peuvent permettre à un observateur de déduire l'existence de ces canaux, tout comme l'analyse des paiements vers ce nœud. C'est d'ailleurs à cause de ces canaux qu'il est impossible de connaître la taille réelle du Lightning Network, puisque seuls les canaux publics peuvent être comptabilisés avec certitude.

Termes associés : **CANAL DE PAIMENT** | **PUBLIC CHANNEL**

PROBLÈME DES GÉNÉRAUX BYZANTINS

ATTAQUE | EN : BYZANTINE GENERALS PROBLEM

Problème formulé pour la première fois par Leslie Lamport, Robert Shostak et Marshall Pease dans le magazine spécialisé *ACM Transactions on Programming Languages and Systems, vol 4, n° 3* « The Byzantine Generals Problem » en juillet 1982. Il est utilisé aujourd'hui pour illustrer les défis en termes de prise de décision quand un système distribué ne peut faire confiance à aucun acteur.

Dans cette métaphore, un groupe de généraux byzantins et leurs armées respectives sont campés autour d'une ville qu'ils souhaitent attaquer ou assiéger. Les généraux sont géographiquement séparés les uns des autres et doivent communiquer par messager pour coordonner leur stratégie. Qu'ils attaquent ou qu'ils battent en retraite n'a pas d'importance, du moment que tous les généraux parviennent à un consensus.

Par conséquent, nous pouvons considérer les exigences suivantes :

- Chaque général doit prendre une décision : attaquer ou battre en retraite (oui ou non) ;
- Une fois la décision prise, elle ne peut plus être modifiée ;
- Tous les généraux doivent se mettre d'accord sur la même décision et l'exécuter de manière synchronisée.

De plus, un général ne peut communiquer avec un autre que par le biais de messages transmis par un coursier, pouvant être retardés, détruits, modifiés ou perdus. Et même si un message réussit à être délivré, un ou plusieurs généraux peuvent choisir (pour une raison quelconque) de trahir la confiance établie et d'envoyer un message frauduleux, et semer le chaos.

Si on applique le dilemme au contexte de la blockchain Bitcoin, chaque général représente un nœud du réseau, devant parvenir à un consensus sur l'état du système. En d'autres termes, la majorité des participants d'un réseau distribué doivent se mettre d'accord et exécuter la même action afin d'éviter une défaillance totale. Le seul moyen de parvenir à un consensus dans ce type de système distribué est d'avoir plus de 2/3 des nœuds de réseau fiables et honnêtes. Donc, si la majorité du réseau décide d'agir de manière malveillante, le système est vulnérable.

Ce dilemme est parfois également appelé « Problème de la diffusion cohérente ».

Termes associés : **CONSENSUS** | **NOEUD**

PROBING

LIGHTNING NETWORK | FR : SONDAGE

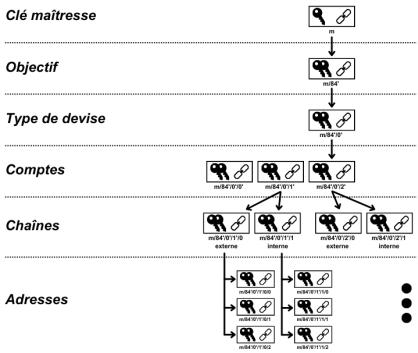
Technique consistant à envoyer des paiements sur le Lightning Network sans intention de les finaliser, dans le but de découvrir des informations sur le réseau, notamment les routes disponibles et les soldes des canaux.

Termes associés : **PAYMENT HASH** | **PREIMAGE**

PROFONDEUR

PORTEFEUILLE | EN : DEPTH

Dans le cadre des portefeuilles HD (déterministes et hiérarchiques), la profondeur désigne le niveau spécifique d'une clé (publique ou privée), d'un code de chaîne, d'une clé étendue ou d'une adresse dans la structure de dérivation du portefeuille depuis la clé maîtresse. Chaque niveau de cette structure peut être vu comme un étage dans un arbre de clés, où la clé maîtresse se trouve à la racine (profondeur 0) et les niveaux suivants définissent divers attributs tels que : l'objectif (profondeur 1), le type de devise (profondeur 2), le compte (profondeur 3), le type de chaîne (profondeur 4), et l'index de l'adresse spécifique (profondeur 5).



Pour passer d'une profondeur à une suivante, on utilise un processus de dérivation depuis une paire de clés parents vers une paire de clés filles.

On parle également parfois d' « étage de dérivation » plutôt que de profondeur.

Termes associés : **CHEMIN DE DÉRIVATION** | **DÉRIVATION**

PROOF-OF-KEY DAY

$ ÉCONOMIE ET RÉGULATION

Initiative annuelle observée chaque 3 janvier (le jour de l'anniversaire du bloc de Genèse) qui encourage les détenteurs de bitcoins à retirer tous en même temps leurs actifs des plateformes d'échange pour les transférer vers des solutions en *self-custody*. L'objectif est de tester la solvabilité des plateformes d'échange, en les forçant à prouver qu'elles détiennent bien les bitcoins de leurs utilisateurs et qu'elles ne font pas de la réserve fractionnaire. Cependant, cette pratique est de moins en moins suivie aujourd'hui, ce qui réduit donc son efficacité. En fin de compte, la meilleure façon de se protéger contre la faillite des *exchanges* reste de conserver ses bitcoins en *self-custody* toute l'année.

Termes associés : **GENÈSE** | **PREUVE DE RÉSERVES**

PROOF OF STAKE

PROTOCOLE | FR : PREUVE D'ENJEU

Mécanisme de consensus alternatif à la preuve de travail, dans lequel les validateurs sont sélectionnés pour proposer et confirmer des blocs en fonction de la quantité de jetons qu'ils immobilisent (« *stake* ») dans le protocole. Plus un participant détient et immobilise de jetons, plus sa probabilité d'être choisi comme validateur augmente.

Ce mécanisme est utilisé par plusieurs systèmes de cryptomonnaies, notamment Ethereum depuis sa transition le 15 septembre 2022. Ses partisans avancent une consommation énergétique bien moindre que celle de la preuve de travail. Ses critiques soulignent qu'il tend à concentrer le pouvoir de validation entre les mains des plus gros détenteurs, reproduisant une dynamique similaire à celle des systèmes financiers traditionnels où la richesse engendre le contrôle dans un système fondamental fermé, car il ne repose pas sur une ressource naturelle externe au protocole.

Bitcoin n'utilise pas la preuve d'enjeu. Son protocole repose exclusivement sur la preuve de travail, qui lie la publication d'un bloc à une dépense énergétique réelle et vérifiable, indépendante de la richesse accumulée par les participants.

Termes associés : **CONSENSUS** | **PREUVE DE TRAVAIL**

PROP - PROPORTIONAL

MINAGE

Méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. PROP répartit simplement la récompense de bloc parmi les mineurs proportionnellement à leur contribution en *shares*. Le calcul des *shares* débute au dernier bloc trouvé par la pool et termine lorsqu'un nouveau bloc est trouvé. Chaque nouveau bloc remet le compteur de *shares* à zéro. Cette méthode de rémunération permet de refléter directement le travail de chacun.

Terme associé : **SHARES**

PROTON WALLET

PORTEFEUILLE

Portefeuille Bitcoin open source en *self-custody* développé par Proton, l'entreprise suisse connue pour ses services axés sur la confidentialité. Disponible en application web et mobile, il est lié à un compte Proton. Sa fonctionnalité distinctive est l'envoi de bitcoins via l'adresse e-mail du destinataire : Proton génère automatiquement une adresse Bitcoin vierge associée au portefeuille de ce dernier.

Terme associé : **PORTEFEUILLE**

PSAN

Ⓢ ÉCONOMIE ET RÉGULATION

Sigle de « Prestataire de Services sur Actifs Numériques ». Statut réglementaire français introduit par la loi PACTE du 22 mai 2019, définissant le cadre juridique applicable aux entreprises qui offrent des services liés aux cryptomonnaies en France. Les services concernés incluent la conservation d'actifs numériques, l'achat et la vente contre monnaie fiat, l'échange entre actifs numériques, et l'exploitation de plateformes de négociation.

Depuis 2019, l'enregistrement obligatoire auprès de l'AMF (Autorité des Marchés Financiers) est requis pour exercer ces activités en France. Un agrément optionnel, plus exigeant, peut également être obtenu. L'enregistrement impose des obligations en matière de lutte contre le blanchiment (LCB-FT) et de vérification d'identité (KYC).

Avec l'entrée en vigueur du règlement européen MiCA (*Markets in Crypto-Assets*), le statut PSAN est progressivement remplacé par le cadre harmonisé européen PSCA (Prestataire de Services sur Crypto-Actifs), applicable dans tous les États membres de l'Union européenne.

Termes associés : **AML - ANTI-MONEY LAUNDERING** | **KYC - KNOW YOUR CUSTOMER**

PSBT

⊶ PORTEFEUILLE

Sigle de « *Partially Signed Bitcoin Transaction* ». C'est une spécification introduite avec le BIP-0174 pour standardiser la manière dont les transactions non finalisées sont construites dans les logiciels en rapport avec Bitcoin, comme les logiciels de portefeuille. Un PSBT encapsule une transaction dans laquelle les inputs peuvent ne pas être entièrement signés. Il inclut toutes les informations nécessaires pour qu'un participant puisse signer la transaction sans nécessiter des données supplémentaires. Ainsi, un PSBT peut se présenter sous 3 formes différentes :

- Une transaction entièrement construite, mais pas encore signée ;
- Une transaction partiellement signée, où certains inputs sont signés tandis que d'autres ne le sont pas encore ;
- Ou une transaction Bitcoin entièrement signée, qui pourra être convertie pour être prête pour la diffusion sur le réseau.

Le format PSBT facilite l'interopérabilité entre différents logiciels de portefeuille et périphériques de signature (hardware wallet). Actuellement, on utilise la version 0 des PSBT, telle que définie dans le BIP-0174, et la version 2 a été définie dans le BIP-0370.

La version 1 des PSBT n'existe pas. Étant donné que la version 0 était la première version, la seconde version était familièrement appelée version 2. Ava Chow, qui a rédigé le BIP-0370, a donc décidé de donner le numéro 2 à cette nouvelle version afin d'éviter toute confusion.

Termes associés : **HARDWARE WALLET** | **MULTISIG**

PSEUDO-ALÉATOIRE

CRYPTOGRAPHIE | EN : PSEUDO-RANDOM

Cet adjectif est employé pour décrire une séquence de nombres qui, bien qu'étant le résultat d'un processus déterministe, affiche des caractéristiques qui se rapprochent de celles idéales d'une séquence véritablement aléatoire. La notion d'aléatoire idéal implique une absence totale de prévisibilité et de corrélation entre les éléments successifs. Un nombre pseudo-aléatoire est généré par un algorithme déterministe et donc, en théorie, il est entièrement prévisible si l'on connaît l'état initial du générateur.

Un générateur de nombres pseudo-aléatoires (« *PRNG* » en anglais, ou « GNPA » en français) est un algorithme utilisé pour produire de tels nombres. Il commence généralement à partir d'une valeur initiale, ou « graine », et applique ensuite une série de transformations mathématiques pour produire la suite de nombres. Du fait de cette déterminabilité, il est important pour la sécurité cryptographique que la graine initiale reste secrète. Les suites pseudo-aléatoires sont largement utilisées dans divers domaines, notamment la cryptographie, car elles manifestent un comportement apparemment aléatoire qui suffit pour de nombreuses applications. L'évaluation de la qualité d'un PRNG repose sur la mesure dans laquelle sa sortie se rapproche d'un véritable aléa en termes de distribution, de corrélations et d'autres propriétés statistiques. Dans le cadre de Bitcoin, les nombres pseudo-aléatoires sont utilisés pour produire des clés privées, ou bien pour produire une graine pour les portefeuilles déterministes et hiérarchiques.

Termes associés : **ENTROPIE** | **GRAINE**

PTLC

LIGHTNING NETWORK

Contrats conditionnels destinés à remplacer les HTLC (*Hash Time Locked Contracts*) pour le routage des paiements sur le Lightning Network. Un PTLC (*Point Time Locked Contract*) fonctionne sur le même principe qu'un HTLC (verrouiller des fonds qui ne sont libérés que si une condition cryptographique est satisfaite) mais utilise des points de courbe elliptique et des signatures au lieu de hashs et de préimages.

Dans un HTLC, tous les sauts d'un même paiement partagent le même hash. Un nœud espion présent sur plusieurs sauts peut corréler les paiements et identifier qu'ils appartiennent à la même transaction. Les PTLC éliminent ce problème : chaque saut utilise une paire clé publique/signature différente, ce qui rend les paiements intraçables d'un saut à l'autre.

Le mécanisme repose sur les *adaptor signatures*, une construction rendue pratique par les signatures de Schnorr. Le receveur fournit un point (clé publique) au payeur. À chaque saut, un nouveau secret est ajouté par agrégation de clés, de sorte que la révélation de la signature finale ne compromet que le secret local, pas la relation entre les sauts. Si un PTLC est publié on-chain, il apparaît comme une dépense Taproot ordinaire, indiscernable d'un paiement classique.

Au-delà de la confidentialité, les PTLC réduisent l'espace de bloc utilisé et

ouvrent la voie à de nouvelles constructions comme les *stuckless payments*, qui permettent de relancer un paiement bloqué sans risque de double paiement. Leur déploiement nécessite des modifications du protocole Lightning et reposait sur la disponibilité des signatures de Schnorr, activées avec Taproot en 2021.

Termes associés : **ADAPTOR SIGNATURE** | **HTLC** | **SCHNORR**

PUBLIC CHANNEL

LIGHTNING NETWORK | FR : CANAL PUBLIC

Canal Lightning annoncé au réseau via un message de *channel announcement* diffusé aux pairs. Contrairement à un canal privé, un canal public est inclus dans le graphe du réseau Lightning : les autres nœuds connaissent son existence, sa capacité et ses politiques de frais, ce qui leur permet de l'utiliser pour router des paiements.

Termes associés : **CANAL DE PAIMENT** | **PRIVATE CHANNEL**

PUBLIC POOL

MINAGE

Pool de minage solo *open source* pour Bitcoin, créée par Benjamin Wilson. Contrairement aux pools classiques qui mutualisent la puissance de calcul et partagent les récompenses, Public Pool fonctionne sur le principe du *solo mining* : chaque mineur tente individuellement de trouver un bloc et, en cas de succès, conserve l'intégralité de la récompense.

La plateforme ne prélève aucun frais sur son instance hébergée. Elle fournit un *template* de bloc aux mineurs, ce qui leur évite de devoir gérer leur propre nœud complet.

Public Pool fait partie de l'écosystème *Open Source Miners United* et s'intègre nativement avec des projets comme le NerdMiner et le Bitaxe.

Termes associés : **MINAGE** | **MINING POOL**

PULL REQUEST

INFORMATIQUE | FR : DEMANDE DE TIRAGE

Dans le cadre de GitHub et d'autres plateformes d'hébergement de code, une *Pull Request* représente une demande faite par un contributeur pour intégrer ses modifications d'une branche à une autre. Elle déclenche une révision de code et une discussion avant que les changements ne soient potentiellement fusionnés (*merge*). Ce processus est très utilisé dans le développement des implémentations de nœuds Bitcoin, notamment Bitcoin Core.

Le terme de « Pull Request » est souvent abrégé par le sigle « PR ».

Termes associés : **GITHUB** | **MERGE**

PURGE DE MEMPOOL

PROTOCOLE

Événement au cours duquel un nœud Bitcoin supprime une partie des transactions en attente dans sa mempool. Chaque nœud maintient en mémoire un ensemble de transactions non confirmées, mais cet espace est limité par des contraintes de ressources et par des règles de politique locale. Lorsque certaines conditions sont réunies, le nœud procède à une purge pour libérer de l'espace.

La cause la plus fréquente est le dépassement de la taille maximale autorisée pour la mempool. Dans Bitcoin Core, cette limite est fixée par défaut à 300 Mo via le paramètre `maxmempool`. Quand ce seuil est atteint, les transactions qui offrent le taux de frais le plus faible sont évincées en priorité afin de conserver uniquement celles qui ont le plus de chances d'être incluses dans un prochain bloc. Une transaction peut également être purgée si elle reste en attente trop longtemps sans être confirmée. Par défaut, cette durée d'expiration est de 336 heures, soit 14 jours, via le paramètre `mempoolexpiry`.

Une transaction purgée n'est pas définitivement annulée. Elle peut être rediffusée sur le réseau par son émetteur ou par un autre nœud qui la conserve encore dans sa propre mempool. La purge est un mécanisme de gestion locale des ressources, et non un rejet définitif par le protocole.

Termes associés : FRAIS DE TRANSACTION | MEMPOOL | NOEUD

Q

QR CODE

‹/› INFORMATIQUE

Type de code-barres en deux dimensions pouvant être scanné rapidement par des appareils mobiles. Le QR code stocke des informations sous forme de pixels noirs et blancs disposés dans un carré sur un fond blanc. Une caractéristique importante des QR codes est qu'ils embarquent des codes de correction d'erreur Reed-Solomon qui permettent de restaurer les données même si le QR code est partiellement obscurci.

On utilise largement le QR code sur Bitcoin pour faciliter la transmission d'informations entre deux appareils. Il est principalement utilisé pour partager les adresses de réception, encodées au format URI conformément au BIP-0021. Mais son usage s'étend aussi aux clés étendues, aux PSBT, aux invoices Lightning, etc.

QR code veut dire « Quick Response Code ».

Terme associé : **URI**

QE - QUANTITATIVE EASING

Ⓢ ÉCONOMIE ET RÉGULATION | FR : ASSOUPLISSEMENT QUANTITATIF

Politique monétaire non conventionnelle par laquelle une banque centrale achète massivement des actifs financiers, principalement des obligations d'État et des titres adossés à des créances, afin d'injecter des liquidités dans le système financier. L'objectif est de faire baisser les taux d'intérêt à long terme, de stimuler le crédit et, in fine, de soutenir l'activité économique lorsque les outils conventionnels (comme la baisse du taux directeur) ont atteint leurs limites, notamment lorsque celui-ci est proche de zéro (*zero lower bound*).

Le QE a été largement déployé par les grandes banques centrales à partir de la crise financière de 2008. La Réserve fédérale américaine (Fed), la Banque centrale européenne (BCE) et la Banque du Japon (BoJ) ont mené des programmes successifs qui ont considérablement gonflé la taille de leurs bilans. Cette expansion monétaire est critiquée pour ses effets redistributifs : en faisant monter le prix des actifs financiers (actions, obligations, immobilier), le QE profite davantage aux détenteurs de ces actifs qu'à l'économie réelle, ce qui illustre l'effet Cantillon.

Dans la communauté Bitcoin, le quantitative easing est souvent cité comme l'une des motivations fondamentales de l'adoption du BTC. Face à une création monétaire perçue comme illimitée et discrétionnaire, Bitcoin offre une politique monétaire transparente et immuable, avec une émission plafonnée à 21 millions d'unités.

Termes associés : **EFFET CANTILLON** | **FIAT** | **INFLATION**

QUBIT

</> INFORMATIQUE

Unité d'information de base sur un ordinateur quantique. Ces qubits peuvent prendre la valeur de 0, la valeur de 1, ou bien une superposition du 0 et du 1. En utilisant cette superposition d'états avec d'autres phénomènes quantiques tels que l'intrication et l'interférence quantique, un ordinateur quantique peut paralléliser les processus de calculs, et donc résoudre certains problèmes spécifiques beaucoup plus rapidement.

Termes associés : **BIT** **CRYPTOGRAPHIE**

R

RACE ATTACK

ATTAQUE | FR : ATTAQUE PAR COURSE

Forme de double dépense qui exploite le délai de propagation des transactions sur le réseau Bitcoin. L'attaquant envoie simultanément deux transactions qui dépensent les mêmes UTXO : l'une vers un commerçant pour payer un bien ou un service, et l'autre vers une adresse qu'il contrôle. Le commerçant, s'il accepte la transaction sans attendre de confirmation dans un bloc, risque de livrer le bien alors que c'est la seconde transaction qui sera finalement confirmée par les mineurs. Cette attaque cible donc exclusivement les paiements acceptés en « zéro confirmation » (*zero-conf*). Pour s'en prémunir, un commerçant peut attendre au moins une confirmation avant de considérer le paiement comme valide, ou utiliser des techniques de détection de doubles dépenses en surveillant activement le réseau.

Termes associés : **DOUBLE DÉPENSE** | **TRANSACTION NON CONFIRMÉE**

RACINE DE MERKLE

CRYPTOGRAPHIE | EN : MERKLE ROOT

Condensat ou « *top hash* » d'un arbre de Merkle, qui représente un résumé de toutes les informations présentes dans l'arbre. Un arbre de Merkle est une structure d'accumulateur cryptographique, parfois également nommée « arbre de hachage ». Dans le cadre de Bitcoin, des arbres de Merkle sont utilisés pour organiser les transactions dans un bloc et pour faciliter la vérification rapide de l'inclusion d'une transaction spécifique. Ainsi, dans les blocs Bitcoin, la racine de Merkle est obtenue en hachant de manière successive les transactions par paires jusqu'à ce qu'il ne reste qu'un seul hachage (la racine de Merkle). Cette dernière est ensuite incluse dans l'entête du bloc correspondant. On retrouve également cet arbre de hachage dans UTREEXO, une structure permettant de condenser l'UTXO set des nœuds, et dans le MAST Taproot.

Terme associé : **ARBRE DE MERKLE**

RANGEPROOFS

CRYPTOGRAPHIE | FR : PREUVES DE PORTÉE

Type de preuve à divulgation nulle de connaissance (*zero-knowledge proof*) utilisé dans les *Confidential Transactions* pour démontrer qu'un montant engagé se situe dans un intervalle valide, sans en révéler la valeur exacte. Les *rangeproofs* empêchent un utilisateur de créer des montants négatifs ou excessivement élevés qui permettraient de générer frauduleusement des unités monétaires, tout en préservant la confidentialité de la transaction.

Dans les premières implémentations des *Confidential Transactions* sur Liquid, les *rangeproofs* reposaient sur des signatures en anneau de Borromée (*Borromean ring signatures*). Ce mécanisme est destiné à être remplacé par les *Bulletproofs++*, qui produisent des preuves plus compactes et plus efficaces à vérifier.

Termes associés : **BULLETPROOFS** | **CONFIDENTIAL TRANSACTIONS**

RAPID GOSSIP SYNC

LIGHTNING NETWORK

Protocole de synchronisation du graphe des canaux Lightning qui permet à un nœud de rattraper l'état du réseau en téléchargeant un instantané compact depuis un serveur, plutôt que de collecter les messages *gossip* un par un via le réseau pair-à-pair.

Un serveur RGS enregistre les annonces et mises à jour de canaux, puis calcule un delta par rapport à la dernière synchronisation du client. Le format omet les signatures et encode les données de manière incrémentale, ce qui réduit considérablement le volume transféré : un instantané complet du graphe occupe environ 3,3 Mo (1,5 Mo compressé), une synchronisation hebdomadaire environ 566 Ko, et une synchronisation quotidienne seulement 99 Ko compressés.

Développé dans le cadre de LDK, ce protocole cible les clients mobiles où la bande passante et le temps de synchronisation sont limités. Le calcul des routes reste côté client, ce qui préserve la vie privée de l'utilisateur.

Termes associés : **GOSSIP** | **LDK - LIGHTNING DEV KIT**

RASPIBLITZ

OUTIL

Projet *open source* qui permet de déployer un nœud complet Bitcoin et Lightning sur un Raspberry Pi. RaspiBlitz fournit un système préconfiguré qui inclut Bitcoin Core, une implémentation Lightning et une suite d'outils complémentaires comme un explorateur de blocs, un serveur Electrum ou des interfaces de gestion du nœud.

Termes associés : **LIGHTNING NETWORK** | **NOEUD**

RAW TRANSACTION

PORTEFEUILLE | FR : TRANSACTION BRUTE / TRANSACTION SÉRIALISÉE

Transaction Bitcoin construite et signée, qui se trouve dans sa forme binaire. Une transaction brute (*raw TX*) est la représentation finale d'une transaction, juste avant qu'elle ne soit diffusée sur le réseau. Cette transaction contient toutes les informations nécessaires à son inclusion dans un bloc :

- La version ;
- Le flag ;
- Les inputs ;
- Les outputs ;
- Le locktime ;
- Le witness.

Ce que l'on appelle « *raw transaction* » représente les données brutes qui sont passées deux fois dans la fonction de hachage `SHA256` pour générer le TXID de la transaction. Pour les transactions SegWit, seule la sérialisation originale (sans le *marker*, le *flag* et le *witness*) est utilisée pour calculer le TXID. Ces données sont ensuite utilisées dans l'arbre de Merkle du bloc pour intégrer la

transaction dans la blockchain.

Ce concept est également parfois nommé « Serialized Transaction ».

Termes associés : **ARBRE DE MERKLE** | **TXID - TRANSACTION IDENTIFIER**

RBF - REPLACE-BY-FEE

PORTEFEUILLE | FR : REMPLACEMENT PAR LES FRAIS

Mécanisme transactionnel permettant à l'expéditeur de remplacer une transaction par une autre avec des frais plus élevés, afin d'accélérer la confirmation de celle-ci. Si une transaction avec des frais trop faibles reste bloquée, l'expéditeur peut utiliser *Replace-By-Fee* (remplacement par les frais) pour augmenter les frais et prioriser sa transaction de remplacement dans les mempools.

RBF est applicable tant que la transaction est dans les mempools ; une fois dans un bloc, elle ne peut plus être remplacée. Lors de l'envoi initial, la transaction doit spécifier sa disponibilité à être remplacée en ajustant la valeur de `nSequence` à une valeur inférieure à `0×fffffffe`. C'est ce que l'on appelle un « flag » RBF. Ce paramètre signale la possibilité de mise à jour de la transaction après sa diffusion, ce qui permet par la suite de faire un RBF. Cependant, il est parfois possible de remplacer une transaction n'ayant pas signalé RBF. Les nœuds utilisant le paramètre de configuration `mempoolfullrbf=1` acceptent ce remplacement même si RBF n'a pas été signalé initialement.

À la différence de CPFP (*Child Pays For Parent*), où c'est le destinataire qui peut agir pour accélérer la transaction, RBF (*Replace-By-Fee*) permet à l'envoyeur de prendre l'initiative d'accélérer sa propre transaction en augmentant les frais.

Termes associés : **CPFP - CHILD PAY FOR PARENT** | **MEMPOOL**

RÉCOMPENSE DE BLOC

PROTOCOLE | EN : BLOCK REWARD

Total des bitcoins récupérés par un mineur lorsqu'il trouve un bloc valide sur Bitcoin. Cette récompense est composée de deux éléments : la subvention de bloc et les frais de transaction. La subvention de bloc est une quantité fixe de bitcoins que le mineur peut créer *ex nihilo*. Cette quantité diminue progressivement au fil des *halvings*. Les frais de transaction sont les frais cumulés payés par les utilisateurs pour effectuer les transactions incluses dans le bloc miné. Les frais sont également des bitcoins créés par le mineur, mais leur quantité est limitée au montant des bitcoins détruits dans les transactions. En effet, les frais d'une transaction représentent la différence entre le total des *inputs* et le total des *outputs*.

La récompense de bloc est distribuée au sein d'une transaction spécifique que l'on appelle « coinbase ». Les bitcoins qui en sont extraits sont automatiquement bloqués durant une période de 100 blocs. C'est ce que l'on appelle la période de maturité.

Termes associés : **COINBASE** | **SUBVENTION DE BLOC**

RÉCURSIF - COVENANT

PROTOCOLE | EN : RECURSIVE - COVENANT

Un covenant récursif sur Bitcoin est un type de contrat intelligent qui impose des conditions non seulement sur la transaction actuelle, mais aussi sur les transactions futures qui dépensent les sorties de cette transaction. Cela permet de créer des chaînes de transactions où chacune doit respecter certaines règles définies par la première de la chaîne. La récursivité crée une séquence de transactions où chacune hérite des restrictions de sa transaction parent. Cela permettrait d'établir un contrôle complexe et à long terme sur la manière dont les bitcoins peuvent être dépensés, mais cela introduirait également des risques au niveau de la liberté de dépense et de la fongibilité.

Pour résumer, un covenant non récursif se limitera uniquement à la transaction qui succède immédiatement à celle qui a établi les règles. Et au contraire, un covenant récursif aura la capacité d'imposer des conditions spécifiques à un bitcoin de manière indéfinie. Les transactions pourront se succéder, mais le bitcoin en question conservera toujours les conditions initiales qui lui sont attachées. Techniquement, l'instauration d'un covenant non récursif intervient lorsque le `scriptPubKey` d'un UTXO définit des restrictions sur le `scriptPubKey` des sorties d'une transaction qui dépense ledit UTXO. En revanche, l'instauration d'un covenant récursif intervient lorsque le `scriptPubKey` d'un UTXO définit des restrictions sur le `scriptPubKey` des sorties d'une transaction qui dépense ledit UTXO, et de tous les `scriptPubKey` qui suivront la dépense de cet UTXO.

De manière plus générale, en informatique, ce que l'on appelle la « récursivité » est la capacité d'une fonction à s'appeler elle-même, ce qui crée une sorte de mise en abyme.

Terme associé : **COVENANT**

REDEEM

RGB

Dans le cadre du protocole RGB, dans une *State Extension*, le terme *Redeem* désigne l'action de récupérer ou d'exploiter une *Valency* préalablement déclarée. La *Valency*, en tant que droit public inscrit dans l'état du contrat, confère à un participant autorisé la possibilité de réclamer une extension précise de cet état. Ce mécanisme permet ainsi de formaliser l'exercice de droits spécifiques définis dans le schéma du contrat, garantissant que seule la partie légitimement habilitée peut initier la transition correspondante et exploiter les ressources associées.

Termes associés : **STATE EXTENSION** | **VALENCY**

REDEEMSCRIPT

SCRIPT

Script qui définit les conditions spécifiques que doivent remplir les inputs pour débloquer les fonds associés à un output P2SH. Dans un UTXO P2SH, le `scriptPubKey` contient le hachage du `redeemScript`. Lorsqu'une transaction souhaite dépenser cet UTXO en entrée, elle doit fournir le `redeemScript` en clair qui correspond au hachage contenu dans le `scriptPubKey`. Le `redeemScript` est donc donné dans le `scriptSig` de l'input, en plus des autres éléments nécessaires pour satisfaire les conditions du script, comme les signatures ou les clés publiques. Cette structure encapsulée garantit que les détails des conditions de dépense restent cachés jusqu'à ce que les bitcoins soient effectivement dépensés. On l'utilise notamment pour les portefeuilles multisignatures Legacy P2SH.

Termes associés : **P2SH** | **SCRIPTSIG**

RÈGLES DE CONSENSUS

PROTOCOLE | EN : CONSENSUS RULES

Règles fondamentales dans Bitcoin, assurant l'intégrité du système en dictant les critères de validation des transactions et des blocs. Ces règles sont partagées et appliquées de manière uniforme par tous les nœuds complets du réseau. Une transaction ou un bloc qui ne respecte pas les règles de consensus est systématiquement rejeté, indépendamment de toute autre considération. Par exemple, une transaction comportant une signature invalide pour l'un de ses *inputs* ne pourra jamais être incluse dans un bloc valide. Un bloc contenant une telle transaction serait lui-même considéré comme invalide par l'ensemble du réseau.

Les règles de consensus se distinguent des règles de politique locale appliquées par chaque nœud (politique de relais, politique de mempool, règles de standardisation). Les règles de politique locales sont configurées individuellement par chaque nœud et peuvent varier de l'un à l'autre, tandis que les règles de consensus sont absolues et identiques pour tous. Ainsi, une transaction valide selon les règles de consensus pourra toujours être incluse dans un bloc, même si elle est jugée non standard par certains nœuds. À l'inverse, une transaction invalide par consensus ne pourra jamais être confirmée, quel que soit le contexte.

Les règles de consensus constituent également la base sur laquelle reposent les soft forks : un soft fork consiste à restreindre les règles de consensus en rendant invalides des transactions ou des blocs qui étaient auparavant valides. Pour faciliter le déploiement d'un soft fork, les règles de standardisation sont souvent utilisées en amont pour rendre non standards les types de transactions qui seront ultérieurement rendus invalides par consensus, ce qui évite qu'un mineur non mis à jour n'inclue ces transactions dans un bloc.

Termes associés : **POLITIQUE DE RELAIS** | **RÈGLES DE STANDARDISATION** | **SOFT FORK**

RÈGLES DE STANDARDISATION

PROTOCOLE | EN : STANDARDNESS RULES

Sous-ensemble de la politique de mempool, constitué des règles adoptées individuellement par chaque nœud Bitcoin, en plus des règles de consensus, pour définir le format accepté des transactions non confirmées prises individuellement. Plus précisément, les règles de standardisation dictent la structure d'une transaction évaluée de manière isolée, indépendamment de son contexte dans la mempool. Ces règles sont configurées et exécutées en local par chaque nœud et peuvent varier d'un nœud à l'autre. Elles s'appliquent exclusivement sur les transactions non confirmées. Ainsi, un nœud n'acceptera une transaction qu'il jugerait non standard que si celle-ci est déjà incluse dans un bloc valide.

Les règles de standardisation existent principalement pour trois raisons :

- Protection contre les attaques DoS : certaines transactions sont plus coûteuses que d'autres à vérifier pour le nœud, et l'asymétrie entre le faible coût d'envoi d'une transaction non confirmée et le coût élevé de son traitement crée un vecteur d'attaque par déni de service ;
- Mécanisme de préparation aux soft forks : rendre non standard un type de transaction avant de le rendre invalide par consensus via un soft fork garantit qu'un mineur non mis à jour n'inclura pas une transaction nouvellement invalide dans un bloc, ce qui rend les soft forks plus sûrs à déployer ;
- Incitation comportementale : les règles de standardisation peuvent orienter les utilisateurs vers des pratiques moins nuisibles pour le réseau, ou décourager légèrement certaines activités, comme le stockage excessif de données arbitraires dans la blockchain.

Notons que la majorité des nœuds laissent les configurations par défaut telles que préétablies dans Bitcoin Core, engendrant de fait une homogénéité des règles de standardisation à travers le réseau. Une transaction qui, bien que conforme aux règles de consensus, ne respecte pas ces règles de standardisation, aura des difficultés à être diffusée sur le réseau. Elle pourra toutefois être incluse dans un bloc valide si jamais elle atteint un mineur. Dans la pratique, ces transactions, qualifiées de « non standard », sont souvent transmises directement à un mineur par des voies externes au réseau pair-à-pair de Bitcoin. C'est souvent le seul moyen pour confirmer ce type de transaction.

Par exemple, une transaction qui n'alloue aucuns frais est à la fois valide selon les règles de consensus et non standard, car la politique par défaut de Bitcoin Core pour le paramètre `minRelayTxFee` est de `0.000001` (en BTC/kvB).

Les règles de standardisation constituent un sous-ensemble de la politique de mempool, elle-même sous-ensemble de la politique de relais des transactions, qui fait partie de la politique de relais au sens large. Il ne faut donc pas confondre les règles de standardisation avec la politique de mempool, qui est un concept plus englobant.

Termes associés : **MEMPOOL** | **POLITIQUE DE MEMPOOL** | **RÈGLES DE CONSENSUS**

REGTEST

PROTOCOLE | FR : TEST DE RÉGRESSION

Environnement de test privé pour Bitcoin permettant aux développeurs de créer un réseau Bitcoin local avec les mêmes règles de base que le *testnet*. Contrairement au *testnet*, un *regtest* est privé, c'est-à-dire qu'il n'est pas possible de s'y connecter librement. Aussi, il offre un contrôle total sur les paramètres du réseau. Il permet notamment le minage rapide de blocs et le test de scénarios spécifiques. Il est utilisé pour le développement, le débogage et l'expérimentation avant la mise en œuvre sur le *testnet* ou le *mainnet*. Les bitcoins en circulation sur un *regtest* n'ont aucune valeur.

Termes associés : **SIGNET** | **TESTNET**

RELAIS

MINAGE | EN : RELAY

Dans le contexte de la preuve de travail sur le réseau Bitcoin, désigne un outil dont la fonction principale est de relayer les nouveaux blocs valides qui ont été trouvés. On distingue parfois le rôle du mineur, qui est impliqué dans l'ensemble du processus de la preuve de travail, depuis la construction du bloc candidat jusqu'à la diffusion du bloc validé, du rôle du hacheur, qui se limite à participer au hachage d'un bloc sans intervenir dans sa sélection ou sa diffusion. Le rôle de relais, quant à lui, consiste à diffuser les nouveaux blocs trouvés par un hacheur. Dans le cadre des pools de minage, ce rôle de relais est assuré par la pool elle-même. C'est ce rôle qui lui procure d'ailleurs beaucoup de pouvoir sur les hacheurs individuels. Le terme de « relayeur » peut aussi désigner la personne ou l'entité qui opère un relais.

Termes associés : **HACHEUR** | **MINING POOL**

RELAIS NOSTR

RÉSEAU | EN : NOSTR RELAY

Serveur qui constitue l'infrastructure de distribution du protocole Nostr. Un relais accepte les événements publiés par les clients via des connexions WebSocket, les stocke (temporairement ou de façon permanente) et les retransmet aux clients abonnés selon des filtres de requête. Contrairement à un serveur centralisé classique, un relais ne peut pas modifier le contenu des événements qu'il transmet, car ceux-ci sont signés cryptographiquement par leurs auteurs. Chaque relais applique ses propres politiques de modération, de stockage et d'accès. Il existe plusieurs types de relais : publics (ouverts à tous), privés (accès restreint), payants (nécessitant un abonnement) et spécialisés (dédiés à certains types de contenu). Un utilisateur Nostr se connecte généralement à plusieurs relais simultanément pour garantir la redondance et la résistance à la censure.

Termes associés : **NOSTR** | **P2P - PAIR-À-PAIR**

RENDEZ-VOUS ROUTING

LIGHTNING NETWORK | FR : ROUTAGE RENDEZ-VOUS

Technique de routage Lightning conçue pour protéger l'identité du destinataire d'un paiement. Dans le routage classique, l'expéditeur construit l'intégralité de la route et connaît donc le nœud final. Avec le *rendez-vous routing*, le destinataire pré-construit un segment de route chiffré depuis un nœud intermédiaire (le *rendez-vous point*) jusqu'à lui-même, et transmet ce segment à l'expéditeur via l'invoice. L'expéditeur complète la route de son côté jusqu'au *rendez-vous point*, sans jamais connaître les nœuds au-delà.

Cette approche présente toutefois des limitations : l'expéditeur ne peut pas insérer ses propres données dans la portion chiffrée par le destinataire, et les montants doivent être fixés à l'avance dans l'oignon partiel, ce qui complique les paiements multi-parties. Pour ces raisons, le *rendez-vous routing* a été supplanté par les *blinded paths*, une technique plus souple qui permet à l'expéditeur de choisir librement les données insérées à chaque saut et qui est réutilisable avec les messages en oignon.

Termes associés : **BLINDED PATHS** | **ROUTAGE EN OIGNON**

REPLACEMENT CYCLING

ATTAQUE

Attaque qui exploite les règles de remplacement de transactions (RBF) pour retirer une transaction non confirmée des mempools des nœuds complets, sans qu'aucune transaction alternative ne prenne sa place. Elle affecte principalement les protocoles multi-parties qui dépendent du CPFP pour augmenter les frais, comme le Lightning Network. Cette vulnérabilité a été divulguée publiquement en octobre 2023 par Antoine Riard (CVE-2023-40231 à CVE-2023-40234).

L'attaque se déroule en plusieurs étapes cycliques. Bob diffuse sa transaction de sortie unilatérale accompagnée d'une transaction enfant pour le *fee bumping* via CPFP. Mallory répond en diffusant sa propre transaction de sortie concurrente, qui entre en conflit avec celle de Bob puisqu'elles dépensent les mêmes entrées, accompagnée d'une transaction enfant aux frais supérieurs. La transaction de Bob est remplacée dans les mempools. Mallory remplace alors sa propre transaction enfant par une transaction sans rapport avec le canal, ce qui provoque l'éviction de sa transaction de sortie. La mempool ne contient alors plus aucune transaction de règlement. Bob peut rediffuser, mais Mallory répète le cycle à chaque tentative, chaque itération lui coûtant un nouvel UTXO ou un taux de frais supérieur.

Cette attaque est particulièrement dangereuse pour les HTLC du Lightning Network, qui doivent être résolus dans une fenêtre temporelle définie. Si l'attaquant empêche la victime de confirmer sa transaction avant l'expiration du délai, il peut s'approprier les fonds. Les implémentations du Lightning Network ont déployé plusieurs mesures d'atténuation : la rediffusion agressive et fréquente des transactions, la surveillance de la mempool locale pour détecter les préimages avant leur éviction, et l'augmentation des délais d'expiration CLTV

(`cltv_expiry_delta`).

Termes associés : **CANAL DE PAIMENT** | **GOSSIP** | **TRANSACTION PINNING**

REPLAY ATTACK

ATTAQUE | FR : ATTAQUE PAR REJEU

Dans le contexte de Bitcoin, une replay attack se produit lorsqu'une transaction valide sur une blockchain est malicieusement reproduite sur une autre blockchain qui dispose du même historique de transactions. Autrement dit, une transaction diffusée sur une chaîne peut être répliquée sur une autre sans le consentement de l'émetteur de la première transaction.

Prenons l'exemple d'un hard fork hypothétique de Bitcoin, nommé « *BitcoinBis* ». Si vous réalisez un paiement en bitcoins pour acheter une baguette chez un boulanger sur la vraie blockchain Bitcoin, ce même boulanger pourrait rejouer cette transaction légitime sur *BitcoinBis* pour obtenir le même montant en cryptomonnaies sur ce fork, sans aucune action de votre part.

Ce type d'attaque peut uniquement survenir en cas d'embranchement de la blockchain avec 2 chaînes indépendantes qui persistent dans le temps. Typiquement, cela peut survenir en cas de hard fork. Pour qu'une attaque par rejeu soit possible, il faut obligatoirement que les 2 blockchains aient un historique commun et que la transaction dupliquée consomme en inputs des UTXOs créés à partir de transactions précédentes qui ont eu lieu avant la scission des deux chaînes, ou à partir de transactions précédentes qui ont elles-mêmes déjà été rejouées lors d'une précédente attaque par rejeu.

De manière générale, dans l'informatique, une replay attack consiste à intercepter et à réutiliser des données valides pour tromper un système en répétant la transmission originale. Cela peut parfois mener à des usurpations d'identité sur un réseau.

Dans le cas d'une replay attack sur une transaction Bitcoin, on parle parfois simplement d'une « transaction replay ».

Termes associés : **HARD FORK** | **UTXO**

RÉSEAU BITCOIN

RÉSEAU | EN : BITCOIN NETWORK

Désigne l'infrastructure globale du système Bitcoin. Le réseau est constitué de l'ensemble des nœuds (ordinateurs) qui exécutent un logiciel implémentant le protocole Bitcoin, et qui se connectent à leurs pairs. Chaque nœud communique en pair-à-pair avec les autres nœuds, afin de télécharger et de vérifier la blockchain, de vérifier et de diffuser les nouveaux blocs, et de vérifier et de diffuser les nouvelles transactions.

Termes associés : **NOEUD** | **P2P - PAIR-À-PAIR**

RÉSERVES FRACTIONNAIRES

ÉCONOMIE ET RÉGULATION	EN : FRACTIONAL RESERVES

Principe de la finance traditionnelle selon lequel les banques ne gardent qu'une fraction des dépôts de leurs clients en tant que réserve disponible, et utilisent le reste pour accorder des prêts et générer des profits. Dans le contexte de Bitcoin, bien que la monnaie elle-même ne soit pas soumise à la création de crédit bancaire, le concept de réserve fractionnaire s'applique lorsqu'une plateforme d'échange conserve seulement une partie des BTC déposés par ses utilisateurs, et utilise le reste pour d'autres fins. Cela pose un risque de liquidité et de solvabilité si tous les utilisateurs décident de retirer leurs fonds simultanément. C'est notamment pour cela qu'il est essentiel de conserver soi-même ses bitcoins en *self-custody*.

Termes associés : **PREUVE DE RÉSERVES** | **SELF-CUSTODY**

RÉSISTANCE AU PARTITIONNEMENT

INFORMATIQUE	EN : PARTITION TOLERANCE

Capacité du réseau Bitcoin à rester unifié et à maintenir le consensus entre les utilisateurs, en maintenant des connexions et en évitant la séparation de certains nœuds du reste du réseau, malgré les tentatives de le fragmenter. Pour qu'un nœud demeure en consensus avec le réseau, il doit maintenir au moins une connexion active avec un ensemble de pairs partageant les mêmes règles de consensus.

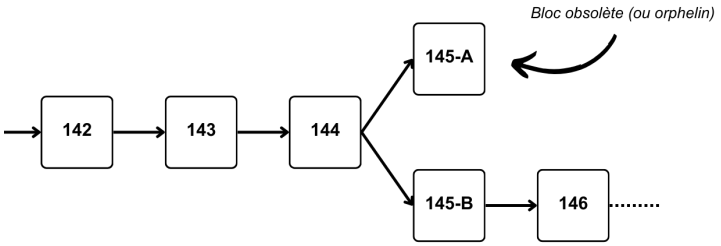
Termes associés : **ECLIPSE** | **NOEUD**

RESYNCHRONISATION

PROTOCOLE	EN : REORGANIZATION

Se réfère à un phénomène dans lequel la blockchain subit une modification de sa structure à cause de l'existence de blocs concurrents à une même hauteur. Cela survient lorsqu'une portion de la chaîne de blocs est remplacée par une autre chaîne ayant une quantité de travail accumulé plus importante.

Ces resynchronisations font partie du fonctionnement naturel de Bitcoin, où différents mineurs peuvent trouver de nouveaux blocs presque simultanément, venant ainsi couper le réseau Bitcoin en deux. Dans de tels cas, le réseau peut se diviser temporairement en chaînes concurrentes. Finalement, lorsque l'une de ces chaînes accumule plus de travail, les autres chaînes sont abandonnées par les nœuds, et leurs blocs deviennent ce que l'on appelle des « blocs obsolètes » ou « blocs orphelins ». Ce processus de remplacement d'une chaîne par une autre est la resynchronisation.



Les resynchronisations peuvent avoir diverses conséquences. Tout d'abord, si un utilisateur avait une transaction confirmée dans un bloc qui s'avère être abandonné, mais que celle-ci ne se retrouve pas dans la chaîne finalement valide, alors sa transaction redevient non confirmée. C'est pour cette raison que l'on vous conseille de toujours attendre au moins 6 confirmations pour considérer une transaction comme réellement immuable. Passé 6 blocs de profondeur, les resynchronisations sont tellement improbables que la chance qu'il y en ait une peut être considérée comme nulle.

Ensuite, au niveau du système global, les resynchronisations impliquent un gaspillage de la puissance de calcul des mineurs. En effet, lorsqu'une division intervient, une partie des mineurs seront sur la chaîne `A`, et une autre partie sur la chaîne `B`. Si la chaîne `B` est finalement abandonnée lors d'une resynchronisation, alors toute la puissance de calcul déployée par les mineurs sur cette chaîne est par définition gaspillée. S'il y a trop de resynchronisations sur le réseau Bitcoin, la sécurité globale de celui-ci est donc réduite. C'est notamment pour cette raison, en partie, que l'augmentation de la taille des blocs ou la réduction de l'intervalle entre chaque bloc (10 minutes) peuvent être dangereuses.

Certains bitcoiners préfèrent parler de « bloc orphelin » pour désigner un bloc périmé. Aussi, même si c'est un anglicisme, on préfère parfois parler dans le langage courant d'une « réorganisation » ou d'une « réorg » plutôt que d'une « resynchronisation ».

Termes associés : **OBSOLÈTE - BLOC** | **PÉRIMÉ - BLOC**

RÉTENTION DE BLOC

✸ ATTAQUE | EN : BLOCK WITHHOLDING

Attaque spécifique au minage dans une pool. C'est une pratique malveillante où un participant de la pool trouve un bloc avec une preuve de travail valide, mais ne le partage pas avec la pool. L'attaquant soumet des preuves de travail partielles (*shares*) pour maintenir l'apparence d'une participation active, mais retient la preuve de travail valide, privant ainsi la pool des récompenses du bloc concerné. Cette tactique vise à diminuer les gains de la pool sans en tirer de bénéfice direct, mais en affectant la rentabilité de celle-ci.

Termes associés : **MINING POOL** | **SELFISH MINING**

RÉUTILISATION D'ADRESSE

CONFIDENTIALITÉ | EN : ADDRESS REUSE

La réutilisation d'adresse se réfère à la pratique d'utiliser une même adresse de réception pour bloquer plusieurs UTXOs, parfois au sein de plusieurs transactions différentes. Les bitcoins sont généralement bloqués à l'aide d'une paire de clés cryptographique qui correspond à une adresse unique. Puisque la blockchain est publique, il est facile de pouvoir consulter quelles adresses sont associées à combien de bitcoins. En cas de réutilisation d'une même adresse pour plusieurs paiements, on peut raisonnablement imaginer que tous les UTXOs associés appartiennent à une même entité. La réutilisation d'adresse pose donc un problème pour la vie privée de l'utilisateur. Elle permet de faire des liens déterministes entre plusieurs transactions et plusieurs UTXOs, ainsi que de perpétuer un traçage de fonds on-chain. Satoshi Nakamoto évoquait déjà ce problème dans son White Paper :

> « *En guise de pare-feu additionnel, une nouvelle paire de clés pourrait être utilisée pour chaque transaction afin de les garder non liées à un propriétaire commun.* » - Nakamoto, S. (2008). « Bitcoin : A Peer-to-Peer Electronic Cash System ». Consulté à l'adresse https://bitcoin.org/bitcoin.pdf.

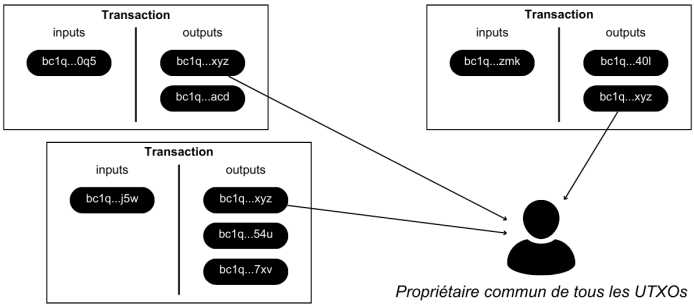
Pour préserver au minimum sa vie privée, il est vivement conseillé de n'utiliser chaque adresse de réception qu'une seule fois. À chaque nouveau paiement, il convient de générer une nouvelle adresse. Pour les outputs de change, il faut également utiliser une adresse vierge. Heureusement, grâce aux portefeuilles déterministes et hiérarchiques, il est devenu très facile d'utiliser une multitude d'adresses. Toutes les paires de clés associées à un portefeuille peuvent être facilement régénérées à partir de la graine. C'est d'ailleurs pour cette raison que les logiciels de portefeuille vous génèrent toujours une nouvelle adresse différente lorsque vous cliquez sur le bouton « Recevoir ».

Termes associés : **ANALYSE DE CHAINE** | **CIOH**

RÉUTILISATION D'ADRESSE - EXTERNE

CONFIDENTIALITÉ | EN : EXTERNAL ADDRESS REUSE

On dit d'une réutilisation d'adresse qu'elle est « externe » lorsqu'elle survient sur plusieurs transactions différentes. Dans cette configuration, la réutilisation d'adresse externe est une heuristique d'analyse de chaîne qui permet d'émettre une hypothèse vraisemblable selon laquelle toutes ces adresses appartiennent à une même entité.

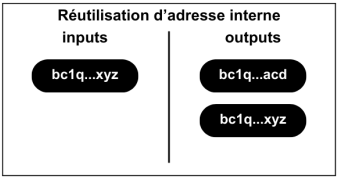


Termes associés : **HEURISTIQUE D'ANALYSE** | **RÉUTILISATION D'ADRESSE**

RÉUTILISATION D'ADRESSE - INTERNE

CONFIDENTIALITÉ | EN : INTERNAL ADDRESS REUSE

On dit d'une réutilisation d'adresse qu'elle est « interne » lorsqu'elle survient au sein d'une même transaction en input et en output. Dans cette configuration, la réutilisation d'adresse interne est une heuristique d'analyse de chaîne qui permet d'émettre une hypothèse vraisemblable sur le change de la transaction. En effet, s'il y a deux outputs et que l'un d'eux utilise la même adresse de réception qu'en input, alors il est vraisemblable que le second output quitte la possession de l'utilisateur initial. L'output avec l'adresse réutilisée représente vraisemblablement le change.



Termes associés : **HEURISTIQUE D'ANALYSE** | **RÉUTILISATION D'ADRESSE**

REVAULT

PORTEFEUILLE

Protocole open source de conservation de bitcoins conçu pour les organisations, développé par Wizardsardine. Revault repose sur un système de *vault* utilisant des transactions présignées et des multisignatures à seuil élevé, sans nécessiter de covenants ni de suppression de clés.

Le protocole formalise l'interaction entre un portefeuille froid et les opérations courantes en permettant la délégation de fonds avec des politiques de restriction : limites de dépense, adresses autorisées, vérifications supplémentaires au-delà d'un certain montant. Chaque participant dispose de ses propres *watchtowers* qui surveillent le réseau et peuvent automatiquement annuler une tentative de dépense non conforme en diffusant des transactions d'urgence présignées. En cas de menace, l'ensemble des fonds peut être déplacé vers un portefeuille

d'urgence par n'importe quel participant individuel.

Le protocole a été conçu à partir de 2019 par Kevin Loaec et Antoine Poinsot et son premier démonstrateur technique a été publié en 2021.

Termes associés : **LIANA** | **VAULT** | **WIZARDSARDINE**

REVOKEANDACK

LIGHTNING NETWORK

Message du protocole Lightning échangé entre deux pairs lors de la mise à jour de l'état d'un canal. Lorsqu'un nouveau paiement est routé ou qu'un HTLC est résolu, les deux parties construisent une nouvelle transaction d'engagement. Le message `revoke_and_ack` remplit deux fonctions simultanées : il révoque l'ancienne transaction d'engagement en transmettant la clé de révocation correspondante, et il accuse réception du nouvel état. Ce mécanisme garantit qu'à tout instant, seul le dernier état du canal est valide. Si un pair tentait de publier un état révoqué, la clé de révocation permettrait à la contrepartie de récupérer l'intégralité des fonds.

Termes associés : **CHANNEL BREACH** | **TRANSACTION D'ENGAGEMENT**

RGB

RGB

Système de contrats intelligents décentralisé et confidentiel, conçu pour fonctionner avec Bitcoin et le Lightning Network. RGB fonctionne sur un modèle de validation côté client et sépare le stockage de l'état des contrats de la blockchain, afin de ne conserver que des engagements cryptographiques sur celle-ci. Ainsi, l'historique complet des états est maintenu en dehors de la chaîne, ce qui permet une meilleure scalabilité et confidentialité. RGB permet ainsi la création de contrats complexes afin d'émettre des tokens, des NFT, des identités décentralisées ou des solutions de DeFi, directement par-dessus Bitcoin.

Sur RGB, la résistance à la double dépense est assurée par l'utilisation de Single-use Seal, un mécanisme cryptographique qui tire parti du fait que les UTXOs sur Bitcoin ne peuvent être utilisés qu'une seule fois. Quant à l'authenticité des jetons, elle est garantie par la vérification côté client de l'historique des états, depuis la création du contrat jusqu'à son état le plus récent.

Termes associés : **CLIENT-SIDE VALIDATION** | **SINGLE-USE SEAL**

RICOCHET

CONFIDENTIALITÉ

Technique consistant à réaliser plusieurs transactions fictives vers soi-même pour simuler un transfert de propriété des bitcoins. Le Ricochet permet d'estomper les spécificités pouvant compromettre la fongibilité d'une pièce Bitcoin. Par exemple, si vous réalisez un coinjoin, votre pièce en sortie de mix sera identifiée comme telle. Cette étiquette de « *pièce issue d'un coinjoin* » peut affecter la fongibilité d'un UTXO. Des entités réglementées, telles que les plateformes d'échange, peuvent refuser d'accepter un UTXO ayant subi un coinjoin, voire exiger des explications de la part de son propriétaire, avec le risque de voir son compte bloqué ou ses fonds gelés. Dans certains cas, la plateforme peut même signaler votre comportement aux autorités étatiques. C'est là que la méthode du Ricochet entre en jeu. Pour estomper l'empreinte laissée par un coinjoin, Ricochet exécute quatre transactions successives où l'utilisateur transfère ses fonds à lui-même sur des adresses différentes. Après cet enchaînement de transactions, l'outil Ricochet achemine finalement les bitcoins vers leur destination finale, comme par exemple une plateforme d'échange. L'objectif est de créer de la distance entre la transaction coinjoin originale et l'acte de dépense final. De cette manière, les outils d'analyse de chaîne vont penser qu'il y a vraisemblablement eu un transfert de propriété après le coinjoin, et qu'il est donc inutile d'entamer des actions à l'encontre de l'émetteur. Le cas d'utilisation le plus courant de Ricochet se présente quand il est nécessaire de dissimuler une participation antérieure à un coinjoin sur un UTXO, notamment pour éviter d'être la cible des politiques LCB/FT des plateformes régulées ou des *blacklists*. L'outil Ricochet est disponible sur le portefeuille Samourai Wallet.

Termes associés : **ANALYSE DE CHAINE** | **COINJOIN**

RIOT PLATFORMS

MINAGE

Entreprise américaine de minage de Bitcoin cotée au NASDAQ sous le symbole RIOT. Fondée en 2000 dans le secteur des biotechnologies, elle a pivoté vers le minage de Bitcoin en 2017 sous le nom de Riot Blockchain, avant d'être renommée Riot Platforms en 2023. L'entreprise possède et exploite des installations de minage au Texas et dans le Kentucky, et figure parmi les plus grandes sociétés de minage de Bitcoin au monde en termes de hashrate. Riot dispose également de capacités d'ingénierie et de fabrication d'infrastructures électriques à travers sa filiale ESS Metron. L'entreprise diversifie ses activités vers l'hébergement de centres de données à haute performance pour des clients tiers.

Terme associé : **MINING POOL**

RIPEMD160

CRYPTOGRAPHIE

Acronyme de *RACE Integrity Primitives Evaluation Message Digest 160*, où « RACE » signifie *Research and development in Advanced Communications technologies in Europe*. C'est une fonction de hachage cryptographique qui génère un condensat de 160 bits à partir d'une entrée arbitraire. Elle est utilisée sur Bitcoin pour transformer une clé publique en une adresse de réception pour les standards Legacy et SegWit v0 (pour SegWit v1, la clé publique n'est pas hachée). Le processus implique d'abord l'application de la fonction de hachage `SHA256` sur la clé publique, suivie de l'application de `RIPEMD160` sur le résultat. Cette combinaison de deux fonctions de hachage distinctes est connue sous le nom de `HASH160` dans le contexte de Bitcoin. `RIPEMD160` est également utilisé dans les portefeuilles déterministes et hiérarchiques pour calculer des empreintes de clés. On utilise notamment `HASH160` pour calculer l'empreinte d'une clé parent, ensuite incluse dans les métadonnées d'une clé étendue (xpub, xprv...).

Terme associé : **FONCTION DE HACHAGE**

RISC ZERO

ORGANISATION

Entreprise et framework open source spécialisés dans la création de preuves à divulgation nulle de connaissance à usage général. RISC Zero fournit une zkVM (*zero-knowledge virtual machine*) basée sur l'architecture de jeu d'instructions RISC-V, permettant d'exécuter du code arbitraire écrit dans des langages courants (Rust, C, C++) et de prouver cryptographiquement la validité de son exécution. Le prouveur génère des preuves STARK, qui sont ensuite encapsulées dans des preuves SNARK via un circuit Groth16 pour réduire leur taille et faciliter la vérification on-chain. Citrea utilise l'infrastructure de RISC Zero pour construire son zkEVM de Type 2, ce qui lui permet de prouver la validité de chaque lot de transactions exécutées sur le rollup. RISC Zero propose également Boundless, un réseau décentralisé de prouveurs permettant d'externaliser la génération de preuves.

Termes associés : **GROTH16** **STARK**

RISQUE DE CONTREPARTIE

ÉCONOMIE ET RÉGULATION | EN : COUNTERPARTY RISK

Risque qu'un tiers auquel on a confié ses fonds ou avec lequel on a conclu un accord ne respecte pas ses obligations, entraînant une perte financière. Dans le contexte de Bitcoin, ce risque se manifeste principalement lorsque des utilisateurs déposent leurs bitcoins sur des plateformes d'échange centralisées (CEX) ou auprès d'autres dépositaires.

Le protocole Bitcoin a été conçu pour éliminer le besoin de confiance envers un tiers. Lorsqu'un utilisateur détient ses propres clés privées, il n'est exposé à aucun risque de contrepartie sur ses bitcoins. En revanche, dès qu'il confie la garde de ses fonds à une entité, il dépend de la solvabilité, de l'honnêteté et de la compétence technique de celle-ci.

L'histoire de Bitcoin est jalonnée de pertes massives liées à ce risque : le piratage de Mt. Gox en 2014, la faillite de QuadrigaCX en 2019 ou l'effondrement de FTX en 2022. La communauté Bitcoin insiste donc sur le principe de *self-custody* : la détention personnelle de ses clés privées reste le seul moyen d'éliminer complètement le risque de contrepartie.

Terme associé : **CUSTODY**

ROAST

CRYPTOGRAPHIE

Protocole de signatures à seuil Schnorr asynchrone, dont l'acronyme signifie *Robust Asynchronous Schnorr Threshold signatures*. ROAST a été proposé en 2022 par Tim Ruffing, Viktoria Ronge, Elliott Jin, Jonas Schneider-Bensch et Dominique Schröder. Il fonctionne comme une surcouche autour d'un schéma de signatures à seuil existant, tel que FROST, pour le rendre résistant aux participants non coopératifs ou aux latences réseau élevées.

Dans un schéma à seuil classique, un sous-ensemble de signataires doit coordonner ses réponses pour produire une signature valide. Si certains participants ne répondent pas ou tentent de perturber le processus, la session de signature peut être bloquée. ROAST résout ce problème en relançant automatiquement des sessions de signature parallèles jusqu'à obtenir un quorum fonctionnel. Ce protocole est particulièrement utile pour les fédérations, comme la fédération du réseau Liquid, où il pourrait permettre d'augmenter considérablement le nombre de signataires tout en maintenant la fiabilité du processus de signature.

Termes associés : **FROST** | **LIQUID NETWORK** | **SCHNORR**

ROBOSATS

OUTIL

Plateforme d'échange *open source* qui permet d'acheter et de vendre des bitcoins en pair-à-pair via le Lightning Network, sans vérification d'identité (KYC). RoboSats fonctionne sur Tor et utilise des identités éphémères (des robots générés automatiquement) pour préserver l'anonymat des utilisateurs. Les échanges reposent sur des *hold invoices* qui servent de mécanisme de séquestre. Les paiements en monnaie fiat se font via des méthodes choisies entre les parties (virement, PayPal, etc.).

Termes associés : **BISQ** | **LIGHTNING NETWORK**

ROLLUP

COUCHE SUPÉRIEURE

Solution de passage à l'échelle de couche 2 (*Layer 2*) qui exécute les transactions en dehors de la chaîne principale, puis publie les données et/ou des preuves de validité sur celle-ci. Le principe consiste à regrouper (*roll up*) de nombreuses transactions en lots traités hors chaîne, tout en ancrant périodiquement un résumé cryptographique sur la couche de base pour hériter de ses garanties de sécurité. Il existe deux grandes familles de rollups. Les *optimistic rollups* supposent les transactions valides par défaut et n'exigent une preuve qu'en cas de contestation (preuve de fraude). Les *ZK rollups* (ou *validity rollups*) publient à chaque lot une preuve cryptographique (SNARK ou STARK) attestant de la validité de toutes les transactions. L'intérêt principal est d'augmenter considérablement le débit transactionnel et de réduire les frais, sans sacrifier la sécurité offerte par la couche de base. Dans le contexte de Bitcoin, des projets comme Citrea ou les Brollups explorent cette approche en utilisant Bitcoin comme couche de disponibilité des données et de règlement final.

Termes associés : **CITREA** | **SURCOUCHE**

ROUND ARK

COUCHE SUPÉRIEURE

Processus périodique au cœur du protocole Ark par lequel l'*Ark Service Provider* (ASP) consolide les transactions VTXO des utilisateurs et les ancre sur la blockchain Bitcoin. Chaque round produit une transaction Ark on-chain contenant un arbre de covenants dans lequel sont inscrits tous les VTXO de sortie créés durant ce cycle.

Le fonctionnement s'apparente conceptuellement à une transaction Bitcoin classique : des VTXO d'entrée sont dépensés et de nouveaux VTXO de sortie sont créés. Cependant, la liquidité on-chain pour les sorties est avancée par l'ASP. En contrepartie, les utilisateurs cèdent leurs anciens VTXO via des *forfeit transactions*. L'ASP récupère cette liquidité ultérieurement, lorsque les VTXO d'entrée expirent. L'atomicité entre la cession des anciens VTXO et la création des nouveaux est garantie par des *connectors*.

Les rounds ont lieu à intervalles réguliers, typiquement toutes les 15 à 60 minutes selon la configuration de l'ASP. Pour les paiements nécessitant un règlement immédiat, les utilisateurs peuvent recourir aux paiements OOR (*out-of-round*) qui contournent ce délai au prix d'un modèle de confiance différent.

Termes associés : **ARK** | **VTXO**

ROUTAGE EN OIGNON

RÉSEAU | EN : ONION ROUTING

Technique de communication qui consiste à encapsuler un message dans plusieurs couches de chiffrement successives, à la manière des pelures d'un oignon. Le message transite par une série de nœuds intermédiaires, et chaque nœud ne peut déchiffrer que la couche qui lui est destinée, révélant uniquement l'identité du nœud suivant. Aucun intermédiaire ne connaît à la fois l'expéditeur et le destinataire final du message.

Ce principe est la base du réseau Tor, où le trafic internet passe par plusieurs relais successifs pour anonymiser les connexions. Sur le Lightning Network, le routage en oignon est utilisé pour acheminer les paiements : chaque nœud de la route ne connaît que le nœud précédent et le nœud suivant, sans pouvoir déterminer l'origine ou la destination du paiement.

Termes associés : **LIGHTNING NETWORK** | **TOR**

RPC - REMOTE PROCEDURE CALL

</> INFORMATIQUE | FR : APPEL DE PROCÉDURE À DISTANCE

Protocole informatique permettant à un programme d'exécuter une procédure sur un autre ordinateur distant, comme si elle était exécutée localement.

Spécifiquement dans le cadre de Bitcoin, on l'utilise pour permettre aux applications d'interagir avec `bitcoind`. Il peut être utilisé pour exécuter des commandes sur un nœud Bitcoin, telles que l'envoi de transactions, la gestion de portefeuilles ou encore l'accès à des informations sur la blockchain. La sécurité de cette interaction est assurée par une authentification via un fichier `.cookie` ou des identifiants, afin que seuls les clients autorisés puissent effectuer des RPC sur le nœud.

Termes associés : **BITCOIN CORE** | **BITCOIND**

RPOW

HISTOIRE

Sigle de « *Reusable Proofs Of Work* ». C'est un système de monnaie électronique par transfert de jetons établis sur des preuves de travail, développé et mis en œuvre par Hal Finney en 2004. RPoW se positionnait comme une amélioration des concepts théoriques de b-money et bit gold.

Contrairement à ces derniers, RPoW a effectivement vu le jour et a été lancé. RPoW aurait pu prendre la place qu'occupe actuellement Bitcoin. C'était le projet le plus abouti de monnaie électronique avant l'invention de Satoshi. Toutefois, Bitcoin surpasse RPoW en résolvant deux problèmes critiques. Premièrement, Bitcoin a introduit un ajustement automatique de la difficulté de minage, un mécanisme absent dans RPoW, évitant ainsi l'inflation due à l'augmentation des capacités de minage et au nombre croissant de mineurs. Deuxièmement, contrairement à la dépendance de RPoW aux serveurs centraux, Bitcoin a instauré un mécanisme de consensus distribué. Ce mécanisme repose sur le principe que les nœuds se synchronisent sur la chaîne avec le plus de travail accumulé, éliminant ainsi la nécessité de serveurs connus. RPoW n'a jamais

reçu le soutien nécessaire pour émerger et être adopté par le grand public.

Contrairement à b-money et bit gold, Satoshi Nakamoto n'a jamais cité RPoW, alors que ce système était sûrement ce qui ressemblait le plus à son invention.

Termes associés : **B-MONEY** | **FINNEY HAL**

RSK

SIDECHAIN

Chaîne latérale (*sidechain*) de Bitcoin conçue pour l'exécution de *smart contracts* compatibles avec Ethereum. Initialement connue sous le nom de Rootstock, cette *sidechain* a été renommée RSK avant de revenir progressivement au nom Rootstock.

Lancée en janvier 2018, RSK fonctionne comme une chaîne latérale rattachée à Bitcoin grâce à un mécanisme de *two-way peg* (ancrage bidirectionnel), qui permet de transférer des bitcoins vers la *sidechain* sous forme de RBTC, des jetons indexés 1 :1 sur le bitcoin.

RSK utilise le *merged mining* : les mineurs de Bitcoin peuvent simultanément sécuriser la *sidechain* RSK sans nécessiter de puissance de calcul supplémentaire. L'objectif est d'étendre les fonctionnalités de Bitcoin par une couche programmable de type Turing-complet, tout en tirant parti de la sécurité de la preuve de travail de Bitcoin.

Termes associés : **MINAGE** | **SIDECHAIN**

RSMPPS

MINAGE

Sigle de « *Recent Shared Maximum Pay Per Share* ». C'est une méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. RSMPPS est similaire à SMPPS, mais avec une priorité accordée aux mineurs ayant contribué récemment.

Cette méthode vise à récompenser les contributions actuelles en augmentant la valeur des shares soumises dans les tours de minage les plus récents, afin de favoriser les mineurs qui restent actifs et fidèles.

Terme associé : **SHARES**

RTL - RIDE THE LIGHTNING

OUTIL

Interface web *open source* permettant de gérer un nœud Lightning via un navigateur. RTL offre un tableau de bord graphique pour les opérations courantes : ouverture et fermeture de canaux, envoi et réception de paiements, gestion de la liquidité, consultation des paiements routés et monitoring du nœud.

RTL est compatible avec les trois principales implémentations de Lightning : LND, Core Lightning (CLN) et Eclair. Il se connecte à l'API du nœud et fournit une interface unifiée quelle que soit l'implémentation sous-jacente.

Cet outil est souvent déployé sur les solutions de *node-in-a-box* comme Umbrel ou Start9, ce qui le rend accessible aux utilisateurs qui ne souhaitent pas interagir en ligne de commande. RTL est développé par Shahana Farooqui et distribué sous licence MIT.

Termes associés : **CANAL DE PAIMENT** | **NOEUD LIGHTNING**

RUNES

COUCHE SUPÉRIEURE

Protocole développé par la même équipe que les Ordinals qui permet l'utilisation de transactions Bitcoin pour graver, frapper et transférer des actifs numériques fongibles autres que le BTC. Contrairement aux inscriptions Ordinals, où chaque unité est unique, chaque rune est identique aux autres, et donc interchangeable, ce qui permet de créer des cryptomonnaies fongibles différentes du BTC sur Bitcoin. Les messages du protocole Runes, appelés *runestones*, sont stockés dans des `OP_RETURN`.

Termes associés : **DIGITAL ARTIFACTS** | **INSCRIPTIONS** | **ORDINALS THEORY**

RUST BITCOIN

OUTIL

Bibliothèque *open source* écrite en Rust pour manipuler les structures de données et les messages du protocole Bitcoin. Elle permet de sérialiser et désérialiser des transactions, des blocs et des messages réseau, de gérer des clés privées, des adresses et la dérivation BIP-0032, ou encore de travailler avec des PSBT.

Rust Bitcoin n'est pas conçue pour la validation de consensus : des écarts, connus et inconnus, existent avec l'implémentation de référence Bitcoin Core, ce qui la rend inadaptée pour un nœud complet. Elle cible plutôt les développeurs qui construisent des portefeuilles, des explorateurs, des outils d'analyse ou tout logiciel qui interagit avec le réseau Bitcoin sans le valider intégralement.

Le projet, publié sous licence CC0 (domaine public), s'accompagne de *crates* compagnons comme `bitcoin_hashes`, `secp256k1` ou `miniscript`, et fonctionne en environnement `no-std` pour les systèmes embarqués.

Termes associés : **BITCOIN CORE** | **SECP256K1**

RUST-LIGHTNING

LIGHTNING NETWORK

Bibliothèque Lightning développée en Rust, créée par Matt Corallo, puis adoptée et développée par Square Crypto (devenu Spiral). Rust-Lightning fournit une implémentation de Lightning. Elle sert de base au *Lightning Development Kit* (LDK).

Terme associé : **LDK - LIGHTNING DEV KIT**

S

STRATEGIC BITCOIN RESERVE

ÉCONOMIE ET RÉGULATION | FR : RÉSERVE STRATÉGIQUE DE BITCOIN

Stock de bitcoins constitué par un État, une institution ou une entreprise comme protection contre l'instabilité économique, la dépréciation monétaire ou les risques géopolitiques.

Termes associés : **ETF** **FIAT**

SAMOURAI WALLET

CONFIDENTIALITÉ

Logiciel de portefeuille Bitcoin pour appareils mobiles Android axé sur la confidentialité. Il offre des fonctionnalités avancées telles que les coinjoins Whirlpool, Stonewall, StonewallX2, Ricochet ou encore Stowaway (payjoin). Samourai implémente également de nombreuses protections pour aider l'utilisateur à protéger sa vie privée face à l'analyse de chaîne.

Le 24 avril 2024, les deux cofondateurs de Samourai Wallet ont été injustement arrêtés pour avoir simplement écrit du code. Je tiens à leur exprimer mon soutien indéfectible. Leur engagement pour la protection de la vie privée et la liberté incarne les valeurs fondamentales de Bitcoin. Ces développeurs n'ont commis aucun crime ; ils ont seulement œuvré pour offrir des outils permettant à chacun de faire valoir ses droits naturels. Afin de les soutenir dans cette épreuve, j'invite chacun à signer la pétition en ligne : https://billandkeonne.org/.

Termes associés : **COINJOIN** **WHIRLPOOL**

SAT - SATOSHI

ÉCONOMIE ET RÉGULATION

Le satoshi, souvent abrégé en « sat », est la plus petite subdivision du bitcoin qui peut être enregistrée sur la blockchain. Il est nommé en l'honneur de l'inventeur de Bitcoin, Satoshi Nakamoto. Un seul bitcoin se divise en 100 000 000 sats, ce qui signifie qu'un satoshi équivaut à 0,00000001 bitcoin. En raison de sa petite valeur unitaire, le sat est souvent utilisé pour établir des prix, en particulier dans les petites transactions. Son utilisation est souvent préférée au BTC sur le Lightning Network.

Termes associés : **BTC** **LIGHTNING NETWORK**

SATOCHIP

ORGANISATION

Entreprise belge spécialisée dans la fabrication de portefeuilles matériels (*hardware wallets*) au format carte à puce. Le produit principal, le Satochip, est une carte NFC équipée d'un élément sécurisé certifié EAL6+, qui permet de stocker des clés privées Bitcoin de manière sécurisée.

La gamme comprend également le SeedKeeper, un dispositif de stockage sécurisé pour les phrases mnémoniques, et le Satodime, un porteur physique de bitcoins. L'ensemble des logiciels associés est open source.

Terme associé : **HARDWARE WALLET**

SATODIME

PORTEFEUILLE

Carte NFC au porteur développée par Satochip, une entreprise belge. Le Satodime prend en charge jusqu'à trois « coffres » indépendants pour stocker des bitcoins. Les clés privées sont générées et scellées dans une puce sécurisée certifiée EAL6+ et ne sont révélées qu'au moment du descellement, ce qui fait que la possession physique de la carte équivaut à la propriété des fonds.

Termes associés : **HARDWARE WALLET** **SATOCHIP**

SATOSHIDICE

HISTOIRE

Site de jeu d'argent en bitcoins, lancé en avril 2012 par Erik Voorhees. Le service fonctionnait comme un jeu de dés probabiliste : les joueurs envoyaient des bitcoins à des adresses qui correspondent à différents niveaux de probabilité de gain, et recevaient automatiquement un paiement en retour selon le résultat. Le site a rapidement dominé l'activité transactionnelle sur la blockchain de Bitcoin. À son apogée en 2012 et début 2013, il représentait près de la moitié des transactions enregistrées sur le réseau.

En juillet 2013, Erik Voorhees a vendu le service pour 126 315 BTC (environ 12,4 millions de dollars à l'époque). En juin 2014, la *Securities and Exchange Commission* (SEC) a engagé des poursuites contre Voorhees pour la vente de titres non enregistrés, liée à des offres de parts de SatoshiDice et de FeedZeBirds à des investisseurs en 2012 et 2013. Il est considéré comme l'un des premiers services commerciaux à grande échelle sur Bitcoin.

Termes associés : **BITCOIN CORE** **SILK ROAD**

SATOSHILABS

ORGANISATION

Entreprise tchèque fondée en 2013 par Marek Palatinus et Pavol Rusnák, connue pour avoir créé le Trezor, le tout premier portefeuille matériel (*hardware wallet*) de l'histoire. Basée à Prague, SatoshiLabs a également contribué à l'élaboration de standards importants pour l'écosystème Bitcoin, dont le BIP-0039 qui définit les phrases mnémoniques.

Termes associés : **HARDWARE WALLET** **TREZOR**

SATSCARD

PORTEFEUILLE

Carte NFC au porteur fabriquée par Coinkite. La Satscard permet de transférer physiquement des bitcoins de main en main sans transaction on-chain au moment de l'échange. Elle contient 10 *slots* indépendants, chacun associé à une clé privée et une adresse unique : les bitcoins sont chargés sur un *slot*, puis « descellés » par le destinataire à l'aide d'un code PIN et d'une application compatible comme Nunchuk.

Termes associés : **COINKITE** **COLDCARD**

SBI CRYPTO

↗ MINAGE

Pool de minage de Bitcoin basée à Tokyo, exploitée par SBI Crypto, filiale du groupe financier japonais SBI Holdings. SBI Holdings pratique le minage en propre depuis 2017, avant d'ouvrir sa pool au public en 2021. Elle s'adresse aussi bien aux mineurs individuels qu'aux opérateurs professionnels.

Terme associé : **MINING POOL**

SCALABILITÉ

✿ PROTOCOLE | EN : SCALABILITY

Fait référence à la capacité de Bitcoin à gérer une augmentation du volume de transactions tout en maintenant des performances acceptables. Bitcoin est confronté à des limitations techniques inhérentes, telles que la taille des blocs, l'intervalle de temps entre chaque bloc, et le fait qu'une transaction ne soit considérée comme immuable qu'après plusieurs confirmations. Ces contraintes empêchent Bitcoin de traiter un nombre illimité de transactions de manière efficace. Lorsque la demande pour l'espace dans les blocs s'intensifie, les utilisateurs sont contraints d'augmenter les frais proposés pour que leurs transactions soient traitées. La scalabilité, c'est-à-dire la capacité de Bitcoin à gérer l'augmentation du volume de transactions, peut être améliorée soit par des modifications directes du protocole, soit par des solutions externes comme le Lightning Network. La question de la scalabilité a toujours été au cœur de vifs débats au sein de la communauté. On retrouve la première tentative d'augmentation de la taille des blocs en octobre 2010 avec un patch proposé par Jeff Garzik. C'est ce sujet de la scalabilité qui a notamment mené à la guerre des blocs entre 2015 et 2017.

Le terme de « scalabilité » est un anglicisme. La bonne traduction du terme anglais « scalability » est « évolutivité » ou bien « passage à l'échelle ». Toutefois, il est généralement admis au sein de la communauté d'utiliser directement le terme de « scalabilité ».

Terme associé : **BLOCKSIZE WAR**

SCB - STATIC CHANNEL BACKUP

ϟ LIGHTNING NETWORK

Mécanisme de sauvegarde introduit par LND permettant de récupérer ses fonds en cas de perte de données d'un nœud Lightning. Le SCB est un fichier (généralement nommé `channel.backup`, à ne pas confondre avec le fichier `channel.db`) qui contient des informations statiques sur les canaux ouverts : identité des pairs, moyens de les contacter et identifiants des canaux.

Contrairement à la base de données `channel.db` qui évolue à chaque paiement, le SCB n'est mis à jour que lors de l'ouverture ou de la fermeture d'un canal. Il ne contient pas l'état détaillé des canaux ni l'historique des transactions. Le SCB ne permet donc pas de restaurer un nœud Lightning en l'état : son rôle est de demander aux pairs de fermer de force les canaux afin de récupérer les fonds *onchain*.

Lors d'une restauration, l'utilisateur fournit sa seed aezeed (pour recréer le portefeuille *onchain*) et le fichier SCB. LND contacte alors chaque pair pour lui signaler la perte de données. Le pair diffuse sa dernière transaction d'engagement pour fermer le canal, et les fonds réapparaissent dans le portefeuille *onchain*. Ce mécanisme dépend de la coopération et de la disponibilité des pairs : si un pair est hors ligne ou non coopératif, les fonds du canal concerné peuvent rester bloqués indéfiniment.

Termes associés : **AEZEED** | **CHANNEL.DB**

SCHEMA

RGB

Dans le cadre du protocole RGB, un Schema désigne un morceau de code déclaratif qui définit l'ensemble des variables, règles et logiques d'affaires (*Business Logic*) régissant le fonctionnement d'un contrat. Il spécifie la structure de l'état, notamment la configuration initiale établie lors de la Genesis, et détaille les types de transitions autorisées ainsi que les conditions strictes de leur validation. Le Schema sert de cadre normatif, garantissant que toute évolution de l'état du contrat se conforme aux règles établies par le développeur du Schema.

Termes associés : **GENESIS** | **STATE TRANSITION**

SCHNORR

CRYPTOGRAPHIE | EN : SCHNORR PROTOCOL

Le protocole de Schnorr est un algorithme de signatures électroniques établi sur la cryptographie sur les courbes elliptiques (ECC). Il est utilisé sur Bitcoin pour dériver une clé publique à partir d'une clé privée et pour signer une transaction avec une clé privée. Sur Bitcoin, tout comme ECDSA, Schnorr est établi sur l'exploitation de la courbe elliptique `secp256k1`, caractérisée par l'équation : $y^2 = x^3 + 7$. Le protocole de signature de Schnorr est implémenté dans le protocole Bitcoin depuis novembre 2021 avec l'activation de la mise à jour Taproot.

Termes associés : **ECDSA** | **TAPROOT**

SCID - SHORT CHANNEL ID

LIGHTNING NETWORK | FR : IDENTIFIANT COURT DE CANAL

Identifiant compact utilisé sur le Lightning Network pour référencer de manière unique un canal de paiement. Plutôt que d'utiliser les clés publiques des deux nœuds ou le txid complet de la transaction de financement, le SCID encode la position exacte du canal dans la blockchain sous un format condensé.

Le SCID est composé de trois éléments concaténés sur 8 octets : le numéro du bloc contenant la transaction de financement (3 octets), l'index de cette transaction dans le bloc (3 octets), et l'index de la sortie (*output*) correspondant au canal (2 octets). Par exemple, un SCID `700000×1×0` signifie : bloc n°700 000, transaction n°1, sortie n°0.

Termes associés : **BOLT-07** | **CANAL DE PAIMENT**

SCORE DE BOLTZMANN

CONFIDENTIALITÉ | EN : BOLTZMANN SCORE

Indicateur utilisé pour évaluer le niveau de confidentialité d'une transaction Bitcoin. Le score de Boltzmann repose sur le nombre d'interprétations possibles d'une transaction, c'est-à-dire le nombre de combinaisons plausibles entre les *inputs* et les *outputs*.

Le principe est inspiré de la physique statistique de Ludwig Boltzmann : plus une transaction admet de combinaisons possibles (de micro-états), plus son entropie est élevée, et plus elle est difficile à analyser. Le score est généralement exprimé en bits d'entropie. Une transaction avec un score de 0 bit n'a qu'une seule interprétation possible (chaque *input* est clairement relié à un *output*). À l'inverse, un *coinjoin* bien construit peut atteindre un score élevé, car les multiples *inputs* et *outputs* de montants identiques génèrent un très grand nombre de combinaisons possibles.

Ce score permet aux utilisateurs d'évaluer objectivement l'efficacité d'un *coinjoin* ou de toute transaction en termes de confidentialité.

Termes associés : **ENTROPIE - ANALYSE DE CHAINE** | **WHIRLPOOL STAT TOOL**

SCORE BASED METHOD

MINAGE

Méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. Ce système de récompense est proportionnel, mais pondéré par le moment auquel la *share* est soumise. La méthode SCORE valorise les parts en fonction du temps écoulé depuis le début du cycle de minage. Plus une part est soumise tardivement dans le cycle, plus sa valeur est élevée. Cette méthode permet d'inciter les mineurs à rester, car à chaque arrêt du minage, le mineur voit son score stagner alors que celui des autres augmente de plus en plus rapidement.

Cette méthode est parfois également nommée « Bitcoin Pooled Mining » (BPM).

Termes associés : **MINING POOL** | **SHARES**

SCRIPT

SCRIPT

Langage de programmation à piles utilisé pour établir des conditions de dépense, et donc, indirectement, sécuriser des bitcoins. Script est essentiellement une liste d'instructions, composée d'opérateurs logiques et de commandes pour manipuler les piles (*stacks*). Il se matérialise par l'utilisation d'opcodes qui donnent des instructions spécifiques qui sont exécutées par les nœuds pour vérifier la validité d'une transaction. Script est un langage non-Turing complet. Il peut être catégorisé comme un langage de niveau intermédiaire (presque bas niveau) inspiré du Forth.

Termes associés : **OPCODES** | **PILE**

SCRIPTLESS SCRIPTS

SCRIPT | FR : SCRIPTS SANS SCRIPT

Concept initialement développé par Andrew Poelstra qui permet l'exécution de contrats intelligents sans exposer explicitement la logique du contrat sur la blockchain Bitcoin. Comme le suggère l'appellation « script sans script », l'idée repose sur l'exécution de scripts (ou de contrats) sans recourir explicitement à des scripts. Ces contrats exploitent les propriétés des signatures de Schnorr qui permettent l'usage des *Adaptor Signatures*, notamment pour réaliser des *Atomic Swaps*. Les conditions du contrat sont appliquées et exécutées off-chain par les parties impliquées, qui sont les seules à en connaître les termes. Contrairement aux contrats intelligents classiques, les *Scriptless Scripts* minimisent leur empreinte sur la blockchain, réduisant ainsi le coût de l'opération. Ces contrats sont aussi plus discrets que les contrats intelligents classiques, qui laissent des traces sur la blockchain. Ils ressemblent donc à des transactions ordinaires, ce qui accroît leur anonset.

Termes associés : **ADAPTOR SIGNATURE** | **SCHNORR**

SCRIPTPUBKEY

SCRIPT

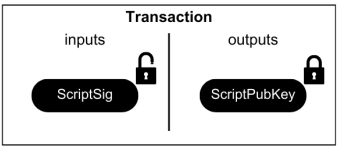
Script situé dans la partie sortie (output) d'une transaction Bitcoin qui définit les conditions sous lesquelles l'UTXO associé peut être dépensé. Ce script permet donc de sécuriser des bitcoins. Dans sa forme la plus courante, le `scriptPubKey` contient une condition qui exige que la prochaine transaction fournisse une preuve de possession de la clé privée correspondant à une adresse Bitcoin spécifiée. C'est souvent réalisé par un script qui demande une signature correspondant à la clé publique associée à l'adresse utilisée pour sécuriser ces fonds. Lorsqu'une transaction tente d'utiliser cet UTXO en entrée (input), elle doit fournir un `scriptSig` qui, une fois associé avec le `scriptPubKey`, satisfait les conditions posées et produit un script valide.

Par exemple, voici un `scriptPubKey` P2PKH classique :

```
OP_DUP
OP_HASH160
OP_PUSHBYTES_20
<hash de la clé publique>
OP_EQUALVERIFY
OP_CHECKSIG
```

Le `scriptSig` correspondant serait :

```
<signature> <clé publique>
```



Pour nommer ce script, on parle également parfois d'un « locking script » ou « script de verrouillage » en français.

Termes associés : **SCRIPTSIG** **UTXO**

SCRIPTSIG

SCRIPT

Élément dans une transaction Bitcoin situé dans les inputs. Le `scriptSig` fournit les données nécessaires pour satisfaire les conditions posées par le `scriptPubKey` de la transaction précédente dont les fonds sont dépensés. Il joue donc un rôle complémentaire au `scriptPubKey`. Typiquement, le `scriptSig` contient une signature numérique et une clé publique. La signature est générée par le propriétaire des bitcoins à l'aide de sa clé privée et prouve qu'il a l'autorisation de dépenser l'UTXO. Dans ce cas, le `scriptSig` démontre que le détenteur de l'input possède la clé privée correspondant à la clé publique associée à l'adresse spécifiée dans le `scriptPubKey` de la transaction sortante précédente.

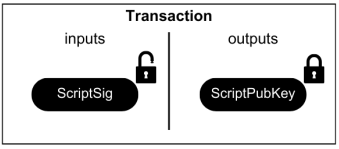
Lorsque la transaction est vérifiée, les données du `scriptSig` sont exécutées dans le `scriptPubKey` correspondant. Si le résultat est valide, cela signifie que les conditions de dépense des fonds ont été remplies. Si toutes les entrées de la transaction fournissent un `scriptSig` qui valide leur `scriptPubKey`, la transaction est valide et pourra être ajoutée à un bloc pour son exécution.

Par exemple, voici un `scriptSig` P2PKH classique :

```
<signature> <clé publique>
```

Le `scriptPubKey` correspondant serait :

```
OP_DUP
OP_HASH160
OP_PUSHBYTES_20
<hash de la clé publique>
OP_EQUALVERIFY
OP_CHECKSIG
```



Le `scriptSig` *est également parfois nommé « unlocking script » ou « script de déverrouillage » en français.*

Termes associés : **SCRIPTPUBKEY** **UTXO**

SCRIPTWITNESS

SCRIPT

Élément dans les entrées de transactions SegWit qui contient les signatures et les clés publiques nécessaires pour déverrouiller les bitcoins envoyés dans la transaction. Semblable au `scriptSig` des transactions Legacy, le `scriptWitness` n'est toutefois pas placé au même endroit. En effet, c'est cette partie, que l'on appelle le « témoin » (« *witness* » en anglais), qui est déplacée dans une structure de données séparée afin de résoudre le problème de la malléabilité des transactions. Chaque input SegWit possède son propre `scriptWitness`, et tous les `scriptWitness` forment ensemble le champ `Witness` de la transaction.

Attention de ne pas confondre le `scriptWitness` *avec le* `witnessScript`*. Tandis que le* `scriptWitness` *contient les données de témoin de tout input SegWit, le* `witnessScript` *définit les conditions de dépense d'un UTXO P2WSH ou P2SH-P2WSH et constitue un script à part entière, à la manière du* `redeemScript` *pour une sortie P2SH.*

Termes associés : **SCRIPTSIG** **SEGWIT**

SDK - SOFTWARE DEVELOPMENT KIT

OUTIL

Ensemble d'outils logiciels fournissant les ressources nécessaires aux développeurs pour créer des applications sur une plateforme spécifique. Un SDK inclut des bibliothèques, des guides de développement, des exemples de code ou encore des processus de compilation. Les SDK facilitent et accélèrent le développement en offrant des modules réutilisables. Sur Bitcoin, il existe le BDK (*Bitcoin Dev Kit*) et le LDK (*Lightning Dev Kit*).

En anglais, les SDK sont également parfois appelés « devkit ».

Termes associés : **BDK - BITCOIN DEV KIT** **LDK - LIGHTNING DEV KIT**

SE - SPARK ENTITY

COUCHE SUPÉRIEURE

Groupe d'opérateurs responsables de la gestion des clés partagées et des opérations de signature au sein du protocole Spark. La SE détient collectivement une moitié de la clé de dépense associée aux UTXO verrouillés dans Spark, l'autre moitié étant détenue par l'utilisateur. Lors d'un transfert sur Spark, la SE participe au renouvellement des parts de clé : elle détruit sa part associée à l'ancien propriétaire et en génère une nouvelle avec le destinataire, ce qui rend l'ancienne part cryptographiquement inutilisable.

La SE n'est pas une entité unique mais un ensemble de *Spark Operators* (SO) indépendants qui coopèrent via un protocole de signature à seuil, basé sur les signatures de Schnorr. Aucun opérateur individuel ne possède la clé complète.

Le modèle de sécurité repose sur une hypothèse 1-parmi-n : il suffit qu'un seul SO soit honnête pour empêcher toute collusion avec un ancien propriétaire. La confiance n'est requise qu'au moment du transfert : une fois les anciennes clés détruites, les opérateurs ne peuvent plus affecter les fonds, même s'ils sont compromis ultérieurement.

Termes associés : **SO - SPARK OPERATOR** | **SPARK**

SEAL DEFINITION

RGB

Dans le cadre du protocole RGB, la Seal Definition constitue la composante d'un Assignment qui relie explicitement un commitment à un UTXO détenu par le nouveau propriétaire. Elle précise l'emplacement de l'état en identifiant l'UTXO auquel celui-ci est associé, ce qui établit de manière formelle la propriété d'un actif ou d'un droit sur RGB.

Termes associés : **ASSIGNMENT** | **SINGLE-USE SEAL**

SECP256K1

CRYPTOGRAPHIE

Nom donné à une courbe elliptique spécifique utilisée dans le cadre du protocole Bitcoin pour l'implémentation des algorithmes de signatures numériques ECDSA (*Elliptic Curve Digital Signature Algorithm*) et Schnorr. La courbe `secp256k1` a été choisie par l'inventeur de Bitcoin, Satoshi Nakamoto. Elle présente certaines propriétés intéressantes, notamment des avantages en termes de performance.

L'utilisation de `secp256k1` sur Bitcoin implique que chaque clé privée (un nombre aléatoire de 256 bits) est mappée à une clé publique correspondante par addition et doublement de point de la clé privée par le point générateur de la courbe `secp256k1`. Cette opération est facile à réaliser dans un sens, mais pratiquement impossible à inverser, ce qui constitue la base de la sécurité des signatures numériques sur Bitcoin.

La courbe `secp256k1` est spécifiée par l'équation de la courbe elliptique $y^2 = x^3 + 7$, ce qui signifie qu'elle a des coefficients a égal à 0 et b égal à 7 dans la forme générale de l'équation d'une courbe elliptique $y^2 = x^3 + ax + b$. `secp256k1` est définie sur un corps fini dont l'ordre est un nombre premier très grand, spécifiquement $p = 2^{256} - 2^{32} - 977$. La courbe a également un ordre de groupe, qui est le nombre de points distincts sur la courbe, un point générateur (ou point G) prédéfini, qui est utilisé dans les opérations de cryptographie pour générer des paires de clés, et un cofacteur qui est égal à 1.

« SEC » désigne « Standards for Efficient Cryptography ». « P256 » désigne le fait que la courbe est définie sur un corps $\mathbf{Z}_p$ où p est un nombre premier de 256 bits. « K » désigne le nom de son inventeur, Neal Koblitz. Enfin, « 1 » désigne que c'est la première version de cette courbe.

Termes associés : **COURBE ELLIPTIQUE** | **ECDSA**

SECP256R1

CRYPTOGRAPHIE

Nom donné à une courbe elliptique définie par le standard NIST pour la cryptographie à clé publique. Elle utilise un champ premier de 256 bits et une équation de courbe elliptique $y^2 = x^3 + ax + b$ avec les constantes :

```
a=-3
```

et

```
b=41058363725152142129326129780047268409114441015993725554835256
314039467401291
```

La courbe `secp256r1` est largement utilisée dans de nombreux protocoles, mais elle n'est pas utilisée dans Bitcoin. En effet, Satoshi Nakamoto a opté pour la courbe `secp256k1`, qui était alors peu connue en 2009. La raison précise de ce choix est inconnue, mais il est possible que ce soit dans le but de minimiser le risque de présence de backdoors. Les paramètres de la courbe $k1$ sont, en effet, nettement plus simples que ceux de la courbe $r1$, en particulier la constante b.

Cette courbe est parfois également nommée « P-256 ».

Termes associés : COURBE ELLIPTIQUE | SECP256K1

SECPOOL

MINAGE

Pool de minage de Bitcoin principalement orientée vers le marché asiatique et russophone. SECPOOL propose des méthodes de rémunération FPPS et PPLNS.

Terme associé : MINING POOL

SECURE ELEMENT

PORTEFEUILLE | FR : ÉLÉMENT SÉCURISÉ

Puce électronique résistante aux attaques physiques, conçue pour stocker des données sensibles et exécuter des opérations cryptographiques dans un environnement isolé. En informatique générale, les *secure elements* sont présents dans les cartes bancaires, les passeports biométriques, les cartes SIM et les modules TPM (*Trusted Platform Module*) des ordinateurs. Ils sont conçus pour résister aux attaques par canaux auxiliaires (*side-channel attacks*), aux attaques par injection de fautes et aux tentatives de rétro-ingénierie physique. Les *secure elements* sont généralement certifiés selon le standard international Critères Communs (CC), avec des niveaux d'assurance allant de EAL1 à EAL7.

Dans le contexte de Bitcoin, les *secure elements* sont utilisés dans les *hardware wallets* pour protéger les clés privées contre tout accès non autorisé, y compris en cas de compromission physique de l'appareil. La clé privée est générée et stockée à l'intérieur de la puce, et les opérations de signature sont réalisées en interne sans jamais exposer la clé à l'extérieur du composant. La plupart des fa-

bricants de *hardware wallets* intègrent un *secure element* dans leurs appareils : Ledger (avec des puces ST de STMicroelectronics), Coldcard (avec des puces ATECC608 de Microchip), BitBox02, Trezor Safe (avec la puce TROPIC01) et d'autres. Certains appareils comme le Jade de Blockstream adoptent une approche alternative dite *virtual secure element*, où la clé est protégée par un schéma de chiffrement distribué plutôt que par une puce physique dédiée.

Terme associé : **PSBT**

SEED NODES

RÉSEAU

Liste statique de nœuds Bitcoin publics, intégrée directement dans le code source de Bitcoin Core (`bitcoin/src/chainparamsseeds.h`). Cette liste sert de points de connexion pour les nouveaux nœuds Bitcoin qui rejoignent le réseau, mais elle n'est utilisée que si les DNS seeds ne fournissent pas de réponse dans les 60 secondes suivant leur sollicitation. Dans ce cas, le nouveau nœud Bitcoin se connectera à ces seed nodes pour établir une première connexion au réseau et demander des adresses IP d'autres nœuds. L'objectif final est d'acquérir les informations nécessaires pour effectuer l'IBD et se synchroniser avec la chaîne qui a le plus de travail accumulé. La liste des seed nodes comprend plusieurs centaines de nœuds, identifiés conformément à la norme établie par le BIP-0155. Ainsi, les seed nodes représentent la troisième méthode de connexion au réseau pour un nœud Bitcoin, après l'éventuelle utilisation du fichier `peers.dat`, créé par le nœud lui-même, et la sollicitation des DNS seeds.

Attention, les seed nodes ne doivent pas être confondus avec les « DNS seeds », qui sont eux la deuxième manière d'établir des connexions.

Terme associé : **DNS SEEDS**

SEEDKEEPER

PORTEFEUILLE

Carte à puce développée par Satochip pour sauvegarder de manière sécurisée des phrases mnémoniques et des mots de passe. Les secrets sont stockés dans un élément sécurisé certifié EAL6+, protégé par un code PIN. Le SeedKeeper peut conserver au moins 50 secrets distincts et permet la duplication vers une seconde carte pour créer une sauvegarde.

Termes associés : **HARDWARE WALLET** **SATOCHIP**

SEEDQR

PORTEFEUILLE

Format de représentation d'une phrase de récupération sous forme de QR code. Développé par le projet SeedSigner, SeedQR encode chaque mot de la phrase mnémonique BIP-0039 par son index dans la liste de mots (position à quatre chiffres, complétée de zéros), puis concatène ces indices en une seule chaîne numérique. Le mode numérique des QR codes permet d'obtenir un résultat compact.

Une variante appelée Compact SeedQR encode directement l'entropie binaire de la phrase (128 bits pour 12 mots, 256 bits pour 24 mots), ce qui réduit la taille du QR code de 35 à 40 %. Le format est adopté par plusieurs signing devices comme SeedSigner, Coldcard Q et Blockstream Jade.

SeedQR offre une alternative à la transcription classique en mots : la phrase peut être sauvegardée sur un support physique (métal, papier) sous forme de QR code, puis scannée directement par le signing device pour importer la *seed* sans saisie manuelle.

Terme associé : **PHRASE DE RÉCUPÉRATION**

SEEDSIGNER

PORTEFEUILLE

Projet *open source* de hardware wallet DIY (à assembler soi-même) basé sur un Raspberry Pi Zero, un écran et une caméra. SeedSigner est conçu pour générer et manipuler des clés privées Bitcoin de manière *air-gapped* et communique exclusivement via des QR codes.

Sa caractéristique principale est sa nature « *stateless* » : l'appareil ne conserve aucune donnée en mémoire entre les utilisations, ce qui signifie que la phrase mnémonique doit être saisie à chaque utilisation, en flashant un QR code. Cette approche élimine le risque d'extraction des clés depuis l'appareil en cas d'accès physique.

Le logiciel est entièrement *open source* et le coût de fabrication est assez faible, ce qui en fait une solution accessible pour signer des transactions ou gérer un multisig. Puisqu'il peut être assemblé à partir de matériel informatique généraliste, c'est également une solution accessible aux personnes vivant sous un régime autoritaire.

Termes associés : **GRAINE** **HARDWARE WALLET**

SEEDXOR

PORTEFEUILLE

Méthode de fractionnement d'une phrase de récupération Bitcoin en plusieurs parties à l'aide de l'opération XOR, introduite par Coinkite pour les portefeuilles matériels Coldcard. Le principe consiste à diviser une phrase mnémonique de 12 ou 24 mots en deux ou plusieurs fragments qui sont eux-mêmes des phrases BIP-0039 valides. Pour reconstituer la phrase originale, toutes les parties doivent être recombinées par XOR (schéma N-de-N, et non M-de-N).

Chaque fragment étant une phrase mnémonique fonctionnelle, il peut servir de portefeuille leurre chargé d'un petit montant, offrant ainsi une forme de déni plausible : un attaquant ne peut pas distinguer un fragment SeedXOR d'une phrase de récupération ordinaire. Le calcul peut être effectué manuellement à l'aide de feuilles de travail imprimables, ou directement sur un Coldcard.

Termes associés : **COLDCARD** | **PHRASE DE RÉCUPÉRATION** | **XOR**

SEGWIT

PROTOCOLE

SegWit, acronyme de « *Segregated Witness* » (Témoin Séparé), est une mise à jour du protocole Bitcoin introduite en août 2017. Elle vise à résoudre plusieurs problèmes techniques, dont la question de la capacité transactionnelle du réseau, le problème de malléabilité des transactions et la facilitation des modifications futures du protocole.

Ce soft fork modifie la structure des transactions en déplaçant les données de signature vers une structure séparée. Concrètement, avec SegWit, les signatures sont retirées de la structure traditionnelle des transactions et insérées dans une structure de données distincte engagée dans un arbre de Merkle séparé au sein du bloc, ce sont les témoins (*witness*). Cette séparation permet d'augmenter la capacité effective de chaque bloc en remplaçant l'ancienne limite de 1 Mo par une nouvelle limite de 4 millions d'unités de poids (4 MWU), tout en restant rétrocompatible avec les anciens nœuds. Cette modification résout également le problème de la malléabilité des transactions. Avant SegWit, les signatures pouvaient être modifiées avant qu'une transaction ne soit confirmée, ce qui changeait l'identifiant de la transaction. Cela rendait difficile la construction de transactions complexes, car une transaction non confirmée pouvait voir son identifiant changer. En séparant les signatures, SegWit rend les transactions non malléables, car tout changement dans les signatures n'affecte plus l'identifiant de la transaction (TXID), mais uniquement l'identifiant du témoin (WTXID). En résolvant le problème de la malléabilité, SegWit a ouvert la voie à d'autres développements en surcouche du système Bitcoin, notamment le réseau Lightning Network, qui permet des transactions rapides et à faible coût.

Termes associés : **MALLÉABILITÉ - TRANSACTION** | **TÉMOIN DE TRANSACTION**

SEGWIT V0

PORTEFEUILLE

Version de script post-SegWit 0. Les scripts SegWit V0 représentent la première famille de scripts introduite avec la mise à jour SegWit de 2017. Les scripts P2WPKH et P2WSH incarnent la version SegWit V0. Les adresses correspondantes commencent toujours par `bc1q` et sont encodées en format `Bech32`.

Termes associés : **P2WPKH** **SEGWIT**

SEGWIT V1

PORTEFEUILLE

Version de script post-SegWit 1. Les scripts SegWit V1 représentent la seconde famille de scripts introduite après la mise à jour SegWit de 2017. En l'occurrence, les scripts SegWit V1 ont été présentés avec la mise à jour Taproot en 2021. Le script P2TR est de la version SegWit V1. Les adresses correspondantes commencent toujours par `bc1p` et sont encodées en format `Bech32m`.

Termes associés : **P2TR** **SEGWIT**

SEGWIT2X

HISTOIRE

Tentative controversée de hard fork visant à doubler la limite de taille des blocs sur Bitcoin, tout en intégrant SegWit. SegWit2x a été introduit lors du New York Agreement en 2017, une réunion confidentielle entre plus de 50 entreprises de l'écosystème qui visait à trouver une solution pour le passage à l'échelle du système. SegWit2x a cherché à augmenter la capacité transactionnelle de Bitcoin en portant la taille maximale d'un bloc à 2 Mo, contre 1 Mo initialement. Le hard fork était prévu au bloc 494 784, soit durant le mois de novembre 2017.

Malgré la mise en place de SegWit en août 2017, le projet 2X de doublement de la taille n'a pas réussi à obtenir un consensus, notamment à cause de la campagne d'opposition « NO2X », ce qui a mené à son annulation au début du mois de novembre. Cet épisode a été perçu par beaucoup d'utilisateurs et de développeurs comme une attaque contre Bitcoin, dans le cadre de la Blocksize War.

SegWit2x est parfois également nommé « B2X » ou « S2X ». Initialement, son nom était « SegWit2Mb ».

Terme associé : **BLOCKSIZE WAR**

SEIGNEURIAGE

$ ÉCONOMIE ET RÉGULATION | EN : SEIGNIORAGE

Revenu que tire un émetteur de monnaie de la différence entre la valeur faciale de la monnaie produite et son coût de production. Historiquement, le seigneuriage désignait le profit réalisé par le seigneur ou le souverain lors de la frappe de pièces métalliques : le coût du métal et de la frappe était inférieur à la valeur nominale de la pièce mise en circulation.

Dans le système monétaire fiat moderne, le seigneuriage prend une ampleur considérable. Les banques centrales créent de la monnaie à un coût quasi nul (par simple écriture comptable ou impression), tandis que cette monnaie circule à sa valeur nominale. Ce mécanisme constitue une forme de taxation implicite sur les détenteurs de monnaie existante, dont le pouvoir d'achat est dilué par chaque nouvelle émission.

Dans Bitcoin, il n'existe pas de seigneuriage au sens traditionnel. L'émission de nouveaux bitcoins via la subvention de bloc ne profite pas à une autorité centrale, mais rémunère les mineurs en contrepartie d'un coût réel : l'énergie et le matériel nécessaires au minage. À terme, la subvention de bloc sera nulle.

Termes associés : **MINAGE** | **SUBVENTION DE BLOC**

SÉLECTION DES PIÈCES

⊸ PORTEFEUILLE | EN : COIN SELECTION

Processus par lequel un logiciel de portefeuille Bitcoin choisit quels UTXOs utiliser comme entrées pour satisfaire les sorties d'une transaction. La méthode de sélection des pièces est importante, car elle a des conséquences sur le coût d'une transaction et la confidentialité de l'utilisateur. Elle vise souvent à minimiser le nombre d'entrées utilisées, afin de réduire la taille de la transaction et les frais associés, tout en tentant de préserver la confidentialité en évitant de fusionner des pièces provenant de sources différentes (CIOH). Plusieurs méthodes existent pour la sélection des pièces, comme le *Knapsack Solver* ou le *Branch-and-Bound*. Lorsque la sélection des pièces est réalisée manuellement par l'utilisateur, on parle alors de « *Coin Control* ».

Termes associés : **COIN CONTROL** | **KNAPSACK SOLVER**

SELF-CUSTODY

⊸ PORTEFEUILLE | FR : GARDE AUTONOME

Désigne la pratique par laquelle un utilisateur garde le contrôle direct de ses clés privées, et donc de ses bitcoins, sans dépendre d'une entité externe pour la gestion de ses actifs.

Termes associés : **CLÉ PRIVÉE** | **HARDWARE WALLET**

SELFISH MINING

ATTAQUE | FR : MINAGE ÉGOÏSTE

Stratégie (ou attaque) dans le minage, où un mineur ou un groupe de mineurs conserve intentionnellement des blocs avec une preuve de travail valide sans les diffuser immédiatement sur le réseau. L'objectif est de conserver une avance sur les autres mineurs en termes de preuve de travail, ce qui leur permet potentiellement de miner plusieurs blocs d'affilée et de les publier en une seule fois, maximisant ainsi leurs gains. Autrement dit, le groupe de mineurs attaquants ne minent pas sur le dernier bloc validé par l'ensemble du réseau, mais plutôt sur un bloc qu'ils ont eux-mêmes créé, qui diffère de celui validé par le réseau.

Ce procédé génère une sorte d'embranchement secret de la blockchain, qui reste inconnu de l'ensemble du réseau jusqu'à ce que cette chaîne alternative dépasse potentiellement la blockchain honnête. Une fois que la chaîne secrète des mineurs attaquants devient plus longue (c'est-à-dire qu'elle contient plus de travail accumulé) que la chaîne honnête et publique, elle est alors diffusée sur l'ensemble du réseau. À ce moment-là, les nœuds du réseau, qui suivent la chaîne avec le plus de travail accumulé, vont se synchroniser sur cette nouvelle chaîne. Il y a donc une réorganisation.

Le *selfish mining* est embêtant pour les utilisateurs, car il diminue la sécurité du système en gaspillant une partie de la puissance de calcul du réseau. En cas de réussite, il conduit également à des réorganisations de la blockchain, affectant ainsi la fiabilité des confirmations de transaction pour les utilisateurs. Cette pratique reste tout de même risquée pour le groupe de mineurs attaquants, car il est souvent plus rentable de miner normalement au-dessus du dernier bloc connu publiquement plutôt que d'allouer de la puissance de calcul à un embranchement secret qui ne dépassera probablement jamais la blockchain honnête. Plus le nombre de blocs dans la réorganisation est grand, plus la probabilité de réussite de l'attaque est basse.

Attention, une attaque par selfish mining ne doit pas être confondue avec une attaque de block withholding (rétention de bloc).

Terme associé : **RÉTENTION DE BLOC**

SENTINEL

PORTEFEUILLE

Application Android de portefeuille *watch-only*, initialement développée par l'équipe de Samourai Wallet et désormais maintenue par la communauté. Sentinel permet de surveiller les soldes et l'historique de transactions d'un portefeuille Bitcoin en important ses clés publiques étendues (`xpub`, `ypub`, `zpub`), sans jamais exposer les clés privées. L'application prend en charge les PSBT, la connexion via Tor et le rattachement à un nœud personnel Dojo.

Termes associés : **ASHIGARU** **SAMOURAI WALLET**

SÉQUENCEUR

COUCHE SUPÉRIEURE | EN : SEQUENCER

Nœud spécialisé responsable de l'ordonnancement des transactions et de la production des blocs dans un rollup. Le séquenceur surveille en continu le mempool, sélectionne les transactions en attente (priorisées par les frais), les exécute contre l'état courant du rollup et produit un nouveau bloc à intervalles réguliers. Il émet ensuite une « confirmation souple » (*soft confirmation*) signée, fournissant une pré-confirmation rapide aux nœuds complets du réseau. Périodiquement, le séquenceur publie des engagements (*commitments*) sur la couche de base (Bitcoin, dans le cas de Citrea) pour ancrer définitivement l'ordre des blocs. Le séquenceur peut influencer l'ordre des transactions et causer des retards de disponibilité (*liveness*), mais ne peut pas falsifier l'état du rollup ni voler des fonds, car les preuves de validité et la disponibilité des données on-chain garantissent l'intégrité des transitions d'état. La décentralisation du séquenceur est un enjeu actif de recherche dans l'écosystème des rollups.

Termes associés : **CITREA** | **ROLLUP**

SETTINGS.JSON

PROTOCOLE

Fichier utilisé dans Bitcoin Core pour stocker les paramètres dynamiques définis via l'interface graphique (GUI) ou via les RPCs. Il conserve des informations sur les configurations de l'utilisateur, telles que les portefeuilles ouverts par exemple. Lorsque Bitcoin Core est redémarré, ce fichier permet de rouvrir automatiquement les portefeuilles qui étaient actifs avant la fermeture de l'application. En cas de fermeture d'un portefeuille via l'interface GUI, il est également supprimé de ce fichier, et il ne sera donc pas rouvert automatiquement au prochain démarrage.

Termes associés : **BITCOIN CORE** | **GUISETTINGS.INI.BAK**

SHA256

CRYPTOGRAPHIE

Sigle pour « *Secure Hash Algorithm 256 bits* ». C'est une fonction de hachage cryptographique produisant un condensat de 256 bits. Conçue par la *National Security Agency* (NSA) au début des années 2000, elle est devenue une norme fédérale pour le traitement des données sensibles. Dans le protocole Bitcoin, la fonction `SHA256` est omniprésente. Elle est employée pour hacher les entêtes des blocs dans le cadre de la preuve de travail. `SHA256` est également utilisée dans le processus de dérivation d'une adresse de réception à partir d'une clé publique. On l'utilise également pour l'agrégation des transactions et des témoins au sein des arbres de Merkle dans les blocs. On retrouve aussi `SHA256` dans le calcul d'empreinte de clés, le calcul de certaines sommes de contrôle et dans de nombreux autres processus autour de Bitcoin. Lorsqu'elle est appliquée deux fois de suite, on parle d'un `HASH256`. Cette double application est celle utilisée majoritairement sur Bitcoin. Lorsque `SHA256` est utilisé conjointement à la fonction `RIPEMD160`, on parle d'un `HASH160`. Ce double hachage est utilisé pour les empreintes de clés et pour le hachage de clés publiques. La

fonction `SHA256` fait partie de la famille des SHA 2.

Terme associé : **FONCTION DE HACHAGE**

SHA512

CRYPTOGRAPHIE

Sigle pour « *Secure Hash Algorithm 512 bits* ». C'est une fonction de hachage cryptographique produisant un condensat de 512 bits. Elle a été conçue par la *National Security Agency* (NSA) au début des années 2000. Pour Bitcoin, la fonction `SHA512` n'est pas utilisée directement dans le cadre du protocole, mais elle est utilisée dans le cadre des dérivations de clés enfants au niveau applicatif, selon le BIP-0032 et le BIP-0039. Dans ces processus, elle est utilisée plusieurs fois dans l'algorithme `HMAC`, ainsi que dans la fonction de dérivation de clés `PBKDF2`. La fonction `SHA512` fait partie de la famille des SHA 2, comme `SHA256`. Son fonctionnement est d'ailleurs très similaire à cette dernière.

Terme associé : **FONCTION DE HACHAGE**

SHAMIR SECRET SHARING - SSS

CRYPTOGRAPHIE | FR : PARTAGE DE SECRET DE SHAMIR

Schéma cryptographique de partage de secret conçu par Adi Shamir en 1979. Il permet de diviser un secret en n parts distinctes, de sorte qu'un minimum de t parts soit nécessaire pour reconstruire le secret original, tandis que t-1 parts ou moins ne révèlent aucune information à son sujet. Le paramétrage (t, n) est défini lors de la création des parts.

Dans l'écosystème Bitcoin, le partage de secret de Shamir est utilisé dans plusieurs contextes. Le protocole Codex32, formalisé dans le BIP-0093, repose sur ce schéma pour permettre la sauvegarde et la reconstruction manuelle de graines. C'est également une option proposée sur les *hardware wallets* Trezor. Les protocoles de signatures à seuil comme FROST s'appuient également sur ce mécanisme lors de la phase de génération distribuée de clés.

Termes associés : **CODEX32** | **FROST** | **SIGNATURE NUMÉRIQUE**

SHARDS - LIGHTNING

LIGHTNING NETWORK | FR : FRACTIONS

Dans le cadre des *Multi-Path Payments (MPP)* ou des *Atomic Multi-Path Payments (AMP)*, un shard est une fraction d'un paiement global. Chaque shard représente une partie du paiement total qui est acheminée séparément via une route différente sur Lightning.

Dans le cadre des MPP, tous les shards partagent le même secret, alors que dans les AMP, chaque shard dispose d'un secret partiel unique. Le destinataire regroupe les shards pour reconstituer et finaliser le paiement complet. Ce mécanisme permet de contourner les limitations de liquidité sur un canal unique.

Terme associé : **MPP - MULTI-PATH PAYMENTS**

SHARDS - RGB

▽ RGB

Dans le cadre du protocole RGB, un *Shard* représente une branche distincte au sein du graphe orienté acyclique (DAG) qui retrace l'historique des *State Transitions* d'un contrat. Il constitue un sous-ensemble cohérent de l'ensemble des transitions, correspondant par exemple à la séquence d'opérations nécessaires pour attester la validité d'un actif particulier depuis la *Genesis*. Ce mécanisme permet d'isoler des segments spécifiques de l'historique global, afin de faciliter la vérification côté client.

Termes associés : **CONSIGNMENT** | **STATE TRANSITION**

SHAREDCOIN

⌀ CONFIDENTIALITÉ

Service de mixage de pièces Bitcoin lancé en 2013 par Blockchain.info, mais qui n'est plus en service aujourd'hui. Ce service proposait aux utilisateurs d'améliorer leur confidentialité sur Bitcoin en combinant leurs transactions avec celles d'autres personnes, grâce à une technique de mixage similaire aux coinjoins. SharedCoin apportait une forme de confidentialité sans nécessiter de faire confiance au coordinateur, car les pièces des utilisateurs restaient sous leur contrôle tout au long du processus. Contrairement aux services de mixage centralisés de l'époque, les bitcoins ne pouvaient pas être volés par l'intermédiaire. SharedCoin a par la suite fait face à des problèmes menant à la désanonymisation de certains de ses mixages au début de l'été 2014.

Terme associé : **COINJOIN**

SHARES

↗ MINAGE | FR : PARTS

Dans le cadre des pools de minage, une share est un indicateur utilisé pour quantifier la contribution d'un mineur individuel au sein de la pool. Cette mesure sert de base pour le calcul de la récompense que la pool redistribue à chaque mineur. Chaque share correspond à un hash qui satisfait une cible de difficulté moins élevée que celle du réseau Bitcoin.

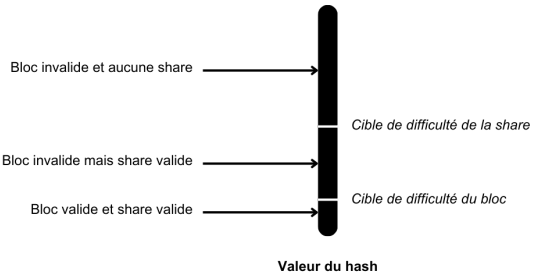
Pour expliquer avec une analogie, considérons un dé à 20 faces. Sur Bitcoin, supposons que la preuve de travail exige de faire un lancer inférieur à 3 pour valider un bloc (c'est-à-dire, obtenir un résultat de 1 ou 2). Maintenant, imaginons qu'au sein d'une pool de minage, la cible de difficulté pour une share est fixée à 10. Ainsi, pour un mineur individuel dans la pool, chaque lancer de dés qui donne un résultat inférieur à 10 compte comme une share valide. Ces shares sont ensuite transmises à la pool par le mineur, afin de réclamer sa rémunération, même si elles ne correspondent pas à un résultat valide pour un bloc sur Bitcoin.

Pour chaque hash calculé, un mineur individuel dans une pool peut rencontrer trois scénarios différents :

- Si la valeur du hash est supérieure ou égale à la cible de difficulté fixée par la pool pour une share, alors ce hash n'est d'aucune utilité. Le mineur modifie

alors son nonce pour tenter un nouveau hash : `hash > share > bloc`.

- Si le hash est inférieur à la cible de difficulté de la share, mais supérieur ou égal à la cible de difficulté de Bitcoin, alors ce hash constitue une share valide qui n'est cependant pas suffisante pour valider un bloc. Le mineur peut envoyer ce hash à sa pool pour réclamer la récompense associée à la share : `share > hash > bloc`.
- Si le hash est inférieur à la cible de difficulté du réseau Bitcoin, il est considéré à la fois comme une share valide et comme un bloc valide. Le mineur transmet ce hash à sa pool, qui s'empresse de le diffuser sur le réseau Bitcoin. Ce hash est également comptabilisé comme une share valide pour le mineur : `share > bloc > hash`.



Ce système de shares est utilisé pour estimer le travail réalisé par chaque mineur individuel au sein d'une pool, sans devoir recalculer individuellement tous les hash générés par un mineur, ce qui serait complètement inefficace pour la pool.

Les pools de minage ajustent la difficulté des shares pour équilibrer la charge de vérification et s'assurer que chaque mineur, qu'il soit petit ou grand, soumet des shares environ toutes les quelques secondes. Cela permet de calculer avec précision le hashrate de chaque mineur et de distribuer les récompenses selon la méthode de calcul de la rémunération choisie (PPS, PPLNS, TIDES...).

Termes associés : DIFFICULTÉ | MINING POOL

SHARES DIFFICULTY

↗ MINAGE | FR : DIFFICULTÉ DES PARTS

Cible de difficulté supérieure à celle de Bitcoin (donc moins difficile), définie par une pool de minage pour qualifier un hash de share et évaluer la contribution de chaque mineur individuel dans la pool. Pour gérer efficacement la charge de vérification et garantir que chaque mineur, quelle que soit sa puissance de calcul, soumette des shares régulièrement, les pools ajustent la difficulté des shares. La difficulté attribuée à chaque mineur détermine le nombre de shares qu'il accumule : par exemple, si un mineur a une difficulté assignée de 10 et qu'il soumet 5 preuves de travail valides à cette difficulté, il obtient 50 shares. Un mineur plus puissant avec une difficulté de 100 qui soumet 7 preuves de travail valides recevra 700 shares. Ce système permet de quantifier précisément le hashrate de chaque mineur et de distribuer les récompenses selon la méthode

de calcul de la rémunération choisie (PPS, PPLNS, TIDES...).

Terme associé : **SHARES**

SHITCOIN

ÉCONOMIE ET RÉGULATION

Expression péjorative fréquemment employée dans le monde des cryptomonnaies pour qualifier des altcoins jugés sans valeur ou sans avenir. Ce terme est utilisé dans deux contextes distincts : dans le milieu général des cryptomonnaies, il désigne un projet qui manque d'innovation technologique, qui ne présente pas de cas d'usage concret, ou qui est principalement le fruit d'une spéculation excessive. Pour les bitcoiners maximalistes, le terme englobe toutes les cryptomonnaies autres que le bitcoin (BTC), faisant ainsi de « shitcoin » un synonyme d'« altcoin ». Le mot est un assemblage de « shit », qui signifie « merde » en anglais, et de « coin » qui signifie « pièce ». L'utilisation initiale du terme remonte à un message posté par l'utilisateur Ribuck le 8 novembre 2010 sur le forum Bitcoin Talk.

Terme associé : **ALTCOIN**

SHOR

INFORMATIQUE | EN : SHOR'S ALGORITHM

Algorithme quantique inventé en 1994 par Peter Shor permettant de factoriser de grands entiers en produit de nombres premiers en temps polynomial. En réduisant le nombre d'opérations nécessaires pour factoriser des entiers, Shor pourrait rendre impraticables les algorithmes de cryptographie établis sur ce problème mathématique comme RSA. Shor peut être légèrement modifié pour agir sur presque tous les algorithmes qui utilisent une structure de groupe. Il dispose notamment déjà d'une variante efficace sur la cryptographie sur les courbes elliptiques (ECDSA, Schnorr...). Shor et ses proches variantes sont donc efficaces sur les algorithmes de cryptographie asymétrique. À l'heure actuelle, nous ne disposons pas encore d'un ordinateur quantique suffisamment puissant et stable pour exécuter avec succès l'algorithme de Shor.

Termes associés : **ECDSA** **SCHNORR**

SIDECAR CHANNEL

LIGHTNING NETWORK

Canal Lightning ouvert par un tiers (partie A) au bénéfice d'un autre utilisateur (partie B), vers un nœud de routage (partie C). Ce mécanisme, introduit dans Lightning Pool par Lightning Labs, permet d'intégrer de nouveaux utilisateurs sur le Lightning Network sans qu'ils aient besoin de détenir du bitcoin au préalable pour financer un canal.

Concrètement, un opérateur de nœud disposant d'un compte Pool financé peut acheter un bail de canal (*channel lease*) au nom d'un tiers. Le nœud de routage vendeur ouvre alors un canal directement vers le bénéficiaire, qui peut immédiatement recevoir des paiements via un pair bien connecté au réseau. Ce modèle crée une source de revenus pour les opérateurs intermédiaires, tout en favorisant la décentralisation du réseau puisque les nouveaux utilisateurs se

connectent à travers une diversité de pairs plutôt qu'un point unique. Les *sidecar channels* permettent aussi de créer des canaux à double financement (*dual-funded*) et peuvent être déployés en tant que *turbo channels*, sans attendre de confirmations on-chain.

Termes associés : **POOL - LIGHTNING** | **TURBO CHANNEL**

SIDECAR TICKET

LIGHTNING NETWORK

Jeton permettant au bénéficiaire d'un *sidecar channel* acheté sur Lightning Pool de revendiquer l'ouverture du canal. Lorsqu'un participant à une enchère Pool achète un canal pour un tiers (par exemple un portefeuille mobile), il génère un *sidecar ticket* contenant les informations nécessaires à l'ouverture. Le bénéficiaire présente ce ticket pour que le vendeur ouvre le canal directement vers lui. Ce mécanisme permet à des utilisateurs qui ne participent pas directement aux enchères Pool de bénéficier de liquidité entrante.

Termes associés : **POOL - LIGHTNING** | **SIDECAR CHANNEL**

SIDECHAIN

SIDECHAIN | FR : CHAÎNE LATÉRALE / PARALLÈLE

Blockchain conçue pour fonctionner en parallèle avec la blockchain principale de Bitcoin. Les deux chaînes sont connectées à l'aide d'un ancrage bilatéral qui permet de faire en sorte que l'actif qui circule sur la sidechain conserve la même valeur que le bitcoin sur la chaîne principale. La sidechain dispose de son propre mécanisme de consensus qui peut être indépendant ou qui peut reposer en partie sur celui de la chaîne principale. Elle permet généralement d'utiliser des fonctionnalités qui ne sont pas disponibles directement sur la chaîne principale, ou bien de bénéficier de fonctionnalités améliorées, comme par exemple : plus de flexibilité dans le développement, des transactions plus rapides et/ou plus confidentielles, ou encore, une capacité transactionnelle plus élevée. Pour ce faire, la sidechain fait des compromis par rapport à la chaîne principale.

Le concept de sidechain a initialement été présenté en 2014 par Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón et Pieter Wuille. Actuellement, les sidechains les plus connues sur Bitcoin sont Liquid et RSK (Rootstock). Ces dernières demeurent toutefois très peu utilisées par rapport à d'autres solutions de surcouche avec un modèle différent comme le Lightning Network.

Termes associés : **ANCRAGE BILATÉRAL** | **LIQUID NETWORK**

SIGHASH_ALL

PROTOCOLE

Type de SigHash Flag utilisé dans les signatures des transactions Bitcoin pour indiquer que la signature s'applique à tous les composants de la transaction. En utilisant `SIGHASH_ALL`, le signataire couvre tous les inputs et tous les outputs. Cela signifie que ni les inputs ni les outputs ne peuvent être modifiés après la signature sans invalider celle-ci. Ce type de SigHash Flag est le plus courant dans les transactions Bitcoin, car il assure une finalité et une intégrité complètes de la transaction.

Termes associés : **SIGHASH FLAG** | **SIGHASH_ALL | SIGHASH_ACP**

SIGHASH_ALL | SIGHASH_ACP

PROTOCOLE

Type de SigHash Flag (`0×81`) combiné avec le modificateur `SIGHASH_ANYONECANPAY` (`SIGHASH_ACP`) utilisé dans les signatures des transactions Bitcoin. Cette combinaison spécifie que la signature s'applique à tous les outputs et uniquement à un input spécifique de la transaction. `SIGHASH_ALL | SIGHASH_ANYONECANPAY` permet à d'autres participants d'ajouter des inputs supplémentaires à la transaction après sa signature initiale. Elle est particulièrement utile dans des scénarios collaboratifs, comme les transactions de financement participatif, où différents contributeurs peuvent ajouter leurs propres inputs tout en préservant l'intégrité des outputs engagés par le signataire initial.

Termes associés : **SIGHASH FLAG** | **SIGHASH_ALL**

SIGHASH_ANYPREVOUT

PROTOCOLE

Proposition d'implémentation d'un nouveau SigHash Flag modificateur dans Bitcoin, introduite avec le BIP-0118. `SIGHASH_ANYPREVOUT` permet une plus grande flexibilité dans la gestion des transactions, en particulier pour des applications avancées comme les canaux de paiement sur le Lightning Network et la mise à jour Eltoo. Le `SIGHASH_ANYPREVOUT` permet de ne lier la signature à aucun UTXO spécifique antérieur (*Any Previous Output*). Utilisé en combinaison avec `SIGHASH_ALL`, il permettrait de signer tous les inputs et tous les outputs d'une transaction, mais sans s'engager sur l'UTXO spécifique dépensé par chaque input. Cela permettrait de réutiliser la signature pour différentes transactions, tant que certaines conditions spécifiées sont remplies.

Ce SigHash modificateur `SIGHASH_ANYPREVOUT` *est hérité de l'idée du* `SIGHASH_NOINPUT` *initialement proposée par Joseph Poon en 2016 pour améliorer son idée du Lightning Network.*

Termes associés : **BIP-0118** | **SIGHASH FLAG**

SIGHASH_ANYPREVOUTANYSCRIPT

PROTOCOLE

Variante du SigHash Flag modificateur `SIGHASH_ANYPREVOUT` dans Bitcoin, introduite dans le BIP-0118. Ce SigHash fonctionne comme `SIGHASH_ANYPREVOUT`, mais en plus de ne pas s'engager sur l'UTXO spécifique dépensé, la signature ne s'engage ni sur le script (scriptCode/tapscript) ni sur le montant de l'input. Cela permet de réutiliser une même signature pour dépenser n'importe quel UTXO utilisant la même clé publique BIP-0118, quel que soit le script ou le montant associé.

Termes associés : **SIGHASH FLAG** | **SIGHASH_ANYPREVOUT**

SIGHASH FLAG

PROTOCOLE

Paramètre dans une transaction Bitcoin permettant de déterminer les composants d'une transaction (inputs et outputs) couvertes par la signature associée, qui deviennent donc immuables. Le SigHash Flag est un octet ajouté à la signature numérique de chaque entrée. Le choix du SigHash Flag affecte donc directement les parties de la transaction qui sont figées par la signature et celles qui peuvent encore être modifiées par la suite. Ce mécanisme assure que les signatures engagent les données de transaction de manière précise et sécurisée, selon l'intention du signataire. Trois principaux SigHash Flags existent :

- `SIGHASH_ALL` (`0×01`) : La signature s'applique à tous les inputs et outputs de la transaction, les verrouillant ainsi intégralement ;
- `SIGHASH_NONE` (`0×02`) : La signature s'applique à tous les inputs, mais à aucun output, permettant la modification des outputs après la signature ;
- `SIGHASH_SINGLE` (`0×03`) : La signature couvre tous les inputs et seulement un output correspondant à l'index de l'input signé.

En complément de ces trois SigHash Flags, le modificateur `SIGHASH_ANYONECANPAY` (`0×80`) peut être combiné avec chacun des types précédents. Quand ce modificateur est utilisé, seul un input est signé, laissant les autres ouverts à modification. Voici les combinaisons existantes avec le modificateur :

- `SIGHASH_ALL | SIGHASH_ANYONECANPAY` (`0×81`) : La signature s'applique à un seul input tout en couvrant tous les outputs de la transaction ;
- `SIGHASH_NONE | SIGHASH_ANYONECANPAY` (`0×82`) : La signature couvre un seul input, sans engager aucun output ;
- `SIGHASH_SINGLE | SIGHASH_ANYONECANPAY` (`0×83`) : La signature s'applique à un seul input et uniquement à l'output ayant le même index que cet input.

Un synonyme parfois utilisé de « SigHash » est « Signature Hash Types ».

Termes associés : **SIGHASH_ALL** | **SIGHASH_NONE - 0X02**

SIGHASH_NONE - 0X02

PROTOCOLE

Type de SigHash Flag utilisé dans les signatures des transactions Bitcoin pour indiquer que la signature s'applique à tous les inputs de la transaction, mais à aucun de ses outputs. L'utilisation de `SIGHASH_NONE` implique que le signataire s'engage uniquement sur les entrées, mais permet que les sorties restent indéterminées ou modifiables après la signature. Ce type de signature est utile dans les cas où le signataire souhaite autoriser d'autres parties à décider de la manière dont les bitcoins seront distribués dans cette transaction.

Termes associés : **SIGHASH FLAG** **SIGHASH_ALL**

SIGHASH_NONE | SIGHASH_ACP

PROTOCOLE

Type de SigHash Flag (`0×82`) combiné avec le modificateur `SIGHASH_ANYONECANPAY` (`SIGHASH_ACP`) utilisé dans les signatures des transactions Bitcoin. Cette combinaison indique que la signature s'applique seulement à un input spécifique, sans engager aucun output. Cela permet aux autres participants de rajouter librement des inputs supplémentaires et de modifier tous les outputs de la transaction.

Termes associés : **SIGHASH FLAG** **SIGHASH_NONE - 0X02**

SIGHASH_SINGLE - 0X03

PROTOCOLE

Type de SigHash Flag utilisé dans les signatures des transactions Bitcoin pour indiquer que la signature s'applique à tous les inputs de la transaction et à un seul output, correspondant à l'index du même input signé. Ainsi, chaque input signé avec `SIGHASH_SINGLE` est lié spécifiquement à un output particulier. Les autres outputs ne sont pas engagés par la signature et peuvent donc être modifiés ultérieurement. Ce type de signature est utile dans des transactions complexes, où les participants veulent lier certains inputs à des outputs spécifiques, tout en laissant de la flexibilité pour les autres outputs de la transaction.

Termes associés : **SIGHASH FLAG** **SIGHASH_NONE - 0X02**

SIGHASH_SINGLE | SIGHASH_ACP

PROTOCOLE

Type de SigHash Flag (`0×83`) combiné avec le modificateur `SIGHASH_ANYONECANPAY` (`SIGHASH_ACP`) utilisé dans les signatures des transactions Bitcoin. Cette combinaison spécifie que la signature s'applique à un seul input spécifique et uniquement à l'output ayant le même index que cet input. Les autres inputs et outputs peuvent être ajoutés ou modifiés par d'autres parties. Cette configuration est utile pour des transactions collaboratives où les participants peuvent ajouter leurs propres inputs pour financer un output spécifique.

Termes associés : **SIGHASH FLAG** **SIGHASH_SINGLE - 0X03**

SIGN-TO-CONTRACT

CRYPTOGRAPHIE

Protocole cryptographique qui permet d'engager des données arbitraires dans le nonce d'une signature numérique, sans que cet engagement soit visible de l'extérieur. Cette technique est l'analogue du Pay-to-Contract, mais appliquée au nonce (la valeur aléatoire k) plutôt qu'à la clé publique. Le principe repose sur le « tweaking » du nonce : à partir d'un nonce original k et d'un engagement c, on calcule un nouveau nonce $k' = k + H(R\|c)$, où $R = k \cdot G$ est le point public du nonce original et H une fonction de hachage. Le nonce public $R' = k' \cdot G$ qui en résulte apparaît comme un point quelconque sur la courbe, mais il encode un engagement vérifiable envers c. Pour prouver l'engagement, il suffit de révéler R et c.

Cette construction trouve plusieurs applications dans Bitcoin. Elle permet notamment d'horodater des documents : l'empreinte d'un fichier est insérée dans le nonce d'une transaction ordinaire. Elle constitue également le mécanisme sous-jacent du protocole anti-exfiltration (anti-exfil), où un logiciel hôte injecte sa propre entropie dans le nonce d'un *hardware wallet* pour empêcher la fuite de la clé privée par canal caché.

Termes associés : **PAY-TO-CONTRACT** | **SCHNORR**

SIGNALING

PROTOCOLE | FR : SIGNALISATION

Mécanisme par lequel les mineurs indiquent leur soutien ou leur préparation à une proposition de soft fork sur Bitcoin. Le signaling s'effectue en positionnant un bit spécifique dans le champ `nVersion` de l'en-tête des blocs qu'ils minent, conformément au processus défini par le BIP-0009 (et ses successeurs comme le BIP-0008).

Chaque bit du champ de version peut être associé à une proposition de soft fork distincte. Lorsqu'un mineur positionne le bit correspondant, il signale qu'il a mis à jour son logiciel et qu'il est prêt à appliquer les nouvelles règles.

Termes associés : **MÉTHODE D'ACTIVATION** | **SOFT FORK**

SIGNATURE AVEUGLE

CRYPTOGRAPHIE | EN : BLIND SIGNATURE

Les signatures aveugles de Chaum sont une forme de signature numérique où l'émetteur d'une signature ne connaît pas le contenu du message qu'il signe. Mais la signature peut ensuite être vérifiée avec le message original. Cette technique a été développée par le cryptographe David Chaum en 1983.

Prenons l'exemple d'une entreprise désirant faire authentifier un document confidentiel, comme un contrat, sans en révéler le contenu. L'entreprise applique un processus de masquage qui transforme cryptographiquement le document original de manière réversible. Ce document modifié est envoyé à une autorité de certification qui appose une signature aveugle sans connaître le contenu sous-jacent. Après avoir reçu le document masqué signé, l'entreprise démasque la signature. Le résultat est un document original authentifié par la

signature de l'autorité, sans que cette dernière ait jamais vu le contenu original.

Les signatures aveugles de Chaum permettent donc de certifier l'authenticité d'un document sans en connaître le contenu, ce qui garantit à la fois la confidentialité des données de l'utilisateur et l'intégrité du document signé.

Sur Bitcoin, on retrouve l'utilisation de ce protocole sur les systèmes de banques chaumiennes en surcouche (Cashu, Fedimint...), mais surtout dans les protocoles de coinjoins chaumiens, afin de s'assurer que le coordinateur ne soit pas en capacité de faire un lien entre un input et un output.

Terme associé : **CHAUMIAN COINJOIN**

SIGNATURE NUMÉRIQUE

CRYPTOGRAPHIE | EN : DIGITAL SIGNATURE

Preuve cryptographique qui démontre la possession d'une clé privée spécifique, associée à une clé publique unique, sans avoir à la divulguer. Sur Bitcoin, on la construit à l'aide de la clé privée et du hash d'une transaction. Elle atteste la propriété des bitcoins concernés et permet de satisfaire un script qui déverrouille un UTXO. Elle est générée grâce à un algorithme de signature numérique sur courbe elliptique tel qu'ECDSA ou le protocole de Schnorr.

Termes associés : **COURBE ELLIPTIQUE** | **ECDSA**

SIGNER À L'AVEUGLE

PORTEFEUILLE | EN : BLIND SIGNING

Processus où un utilisateur signe une transaction Bitcoin sans pouvoir vérifier précisément les détails de ce qu'il signe. Dans le cadre des *hardware wallets*, signer à l'aveugle est dangereux, car l'utilisateur doit se fier à l'ordinateur connecté pour afficher les informations (destinataires, frais, montants...). Cela l'expose à des vulnérabilités, car l'ordinateur est connecté à internet et dispose d'une surface d'attaque beaucoup plus grande que le périphérique de signature. Si l'ordinateur est compromis, il peut manipuler les détails affichés et envoyer une transaction malveillante à signer sans que l'utilisateur en ait conscience. Pour éviter ce risque, il est recommandé d'utiliser des *hardware wallets* avec un écran intégré, et de vérifier les informations sur celui-ci avant de signer les transactions.

Termes associés : **HARDWARE WALLET** | **SIGNATURE NUMÉRIQUE**

SIGNET

PROTOCOLE

Versions spécifiques du réseau Bitcoin conçues pour le développement et les tests. Les signets simulent le comportement du réseau principal (*mainnet*) mais avec la possibilité de contrôler divers paramètres. Ils offrent ainsi un environnement pour tester de nouvelles fonctionnalités ou modifications sans risquer de perturber le réseau principal et sans en subir les frais. Par rapport au *testnet*, les signets offrent un contrôle plus structuré sur la génération de blocs, souvent gérée par une ou plusieurs entités de confiance ou par un mécanisme de consensus personnalisé. Cela permet de créer des scénarios de test plus prévisibles, par rapport au *testnet* qui subit les aléas du minage, de la même manière que le *mainnet*.

Termes associés : **MAINNET** **TESTNET**

SIGOPS

PROTOCOLE | FR : OPÉRATIONS DE SIGNATURE

Désigne les opérations de signature numérique nécessaires pour valider les transactions. Chaque transaction Bitcoin peut contenir plusieurs inputs, chacun pouvant nécessiter une ou plusieurs signatures pour être considéré comme valide. La vérification de ces signatures se fait grâce à l'utilisation d'opcodes spécifiques que l'on nomme les « sigops ». Concrètement, cela inclut `OP_CHECKSIG`, `OP_CHECKSIGVERIFY`, `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY`. Ces opérations font peser une certaine charge de travail sur les nœuds du réseau qui doivent les vérifier. Pour éviter des attaques DoS par inflation artificielle du nombre de sigops, le protocole impose donc une limite sur le nombre de sigops autorisées par bloc, afin de garantir que la charge de validation reste gérable pour les nœuds. Cette limite est actuellement de 80 000 sigops maximum par bloc. Pour compter, les nœuds suivent les règles suivantes :

Dans le `scriptPubKey`, `OP_CHECKSIG` et `OP_CHECKSIGVERIFY` comptent pour 4 sigops. Les opcodes `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY` comptent pour 80 sigops. En effet, lors du comptage, ces opérations sont multipliées par 4 lorsqu'elles ne font pas partie d'un input SegWit (pour un P2WPKH, le nombre de sigops sera donc de 1) ;

Dans le `redeemScript`, les opcodes `OP_CHECKSIG` et `OP_CHECKSIGVERIFY` valent également 4 sigops, `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY` sont comptabilisés pour `4n` s'ils précèdent `OP_n`, ou 80 sigops dans le cas contraire ;

Pour le `witnessScript`, `OP_CHECKSIG` et `OP_CHECKSIGVERIFY` valent 1 sigop, `OP_CHECKMULTISIG` et `OP_CHECKMULTISIGVERIFY` sont comptés pour `n` s'ils sont introduits par `OP_n`, ou 20 sigops autrement ;

Dans les scripts Taproot, les sigops sont traitées de manière différente par rapport aux scripts traditionnels. Au lieu de compter directement chaque opération de signature, Taproot introduit un budget de sigops pour chaque entrée de transaction, qui est proportionnel à la taille de cette entrée. Ce budget est de 50 sigops plus la taille en octets du témoin de l'input. Chaque opération de signature

réduit ce budget de 50. Si l'exécution d'une opération de signature fait chuter le budget en dessous de zéro, le script est invalide. Cette méthode permet plus de flexibilité dans les scripts Taproot, tout en maintenant une protection contre les abus potentiels liés aux sigops, en les liant directement au poids de l'entrée. Ainsi, les scripts Taproot ne sont pas pris en compte dans la limite des 80 000 sigops par bloc.

Termes associés : **SIGNATURE NUMÉRIQUE** | **TAPROOT**

SILENT PAYMENTS

CONFIDENTIALITÉ | FR : PAIEMENTS SILENCIEUX

Méthode pour utiliser des adresses Bitcoin statiques afin de recevoir des paiements sans pour autant produire de réutilisation d'adresse, sans interaction et sans lien visible on-chain entre les différents paiements et l'adresse statique. Cette technique élimine le besoin de générer de nouvelles adresses de réception vierges pour chaque transaction, ce qui permet d'éviter les interactions habituelles dans Bitcoin où le destinataire doit fournir une nouvelle adresse au payeur.

Avec les Silent Payments, le payeur utilise les clés publiques du destinataire (clé publique de dépense et clé publique de scan) et la somme de ses clés privées personnelles en input pour générer une adresse vierge pour chaque paiement. Seul le destinataire, grâce à ses clés privées, peut calculer la clé privée correspondant à cette adresse de paiement. On utilise ECDH (*Elliptic-Curve Diffie-Hellman*), un algorithme cryptographique d'échange de clés, pour établir un secret partagé qui sert ensuite à dériver l'adresse de réception et la clé privée (uniquement du côté du destinataire). Pour identifier les Silent Payments qui leur sont destinés, les destinataires doivent scanner la blockchain et examiner chaque transaction correspondant aux critères du protocole. Contrairement au BIP-0047, qui utilise une transaction de notification pour établir le canal de paiement, les Silent Payments suppriment cette étape, ce qui permet d'économiser une transaction. Toutefois, le compromis est que le destinataire doit scanner l'ensemble des transactions potentielles pour déterminer, en appliquant ECDH, si elles lui sont adressées.

Par exemple, l'adresse statique de Bob B représente la concaténation de sa clé publique de scan et de sa clé publique de dépense :

$$B = B_{scan} \parallel B_{spend}$$

Ces clés sont simplement dérivées à partir du chemin suivant :

```
scan : m / 352' / 0' / 0' / 1' / 0
spend : m / 352' / 0' / 0' / 0' / 0
```

Cette adresse statique est publiée par Bob. Alice la récupère pour faire un Silent Payment vers Bob. Elle calcule l'adresse de paiement P_0 appartenant à Bob de cette façon :

$$P_0 = B_{spend} + hash(inputHash \cdot a \cdot B_{scan} \parallel 0) \cdot G$$

Où :

$$inputHash \ = \ hash(outpoint_L \parallel A)$$

Avec :

- B_{scan} : La clé publique de scan de Bob (adresse statique) ;
- B_{spend} : La clé publique de dépense de Bob (adresse statique) ;
- A : La somme des clés publiques en input (tweak) ;
- a : La clé privée du tweak, c'est-à-dire la somme de toutes les paires de clés utilisées dans les `scriptPubKey` des UTXOs consommés en inputs de la transaction d'Alice ;
- $outpoint_L$: Le plus petit outpoint (lexicographiquement) utilisé en input de la transaction d'Alice ;
- $\parallel$: La concaténation (opération qui consiste à mettre bout-à-bout les opérandes) ;
- G : Le point générateur de la courbe elliptique `secp256k1` ;
- $hash$: La fonction de hachage SHA256 taguée avec `BIP0352/SharedSecret` ;
- P_0 : La première clé publique / adresse unique pour le paiement vers Bob ;
- 0 : Un entier permettant de générer plusieurs adresses de paiement uniques.

Bob scanne la blockchain pour trouver son Silent Payment de cette manière :

$$P_0 \ = \ B_{spend} \ + \ hash(inputHash \cdot b_{scan} \cdot A \parallel 0) \cdot G$$

Avec :

- b_{scan} : La clé privée de scan de Bob.

S'il trouve P_0 comme une adresse qui contient un Silent Payment lui étant adressé, il calcule p_0, la clé privée permettant de dépenser les fonds envoyés par Alice sur P_0 :

$$p_0 \ = \ (b_{spend} \ + \ hash(inputHash \cdot b_{scan} \cdot A \parallel 0)) \bmod n$$

Avec :

- b_{spend} : La clé privée de dépense de Bob ;
- n : l'ordre de la courbe elliptique `secp256k1`.

En plus de cette version de base, on peut également utiliser des labels qui permettent de générer plusieurs adresses statiques différentes à partir d'une même adresse statique de base, dans le but de séparer plusieurs utilisations, sans pour autant multiplier irraisonnablement le travail requis lors du scanning.

Terme associé : LABEL - SILENT PAYMENTS

SILK ROAD

ORGANISATION

Silk Road était une plateforme du *dark web* fondée par Ross Ulbricht, également connu sous le pseudonyme de *Dread Pirate Roberts*. Lancée en 2011, cette place de marché en ligne permettait aux utilisateurs d'acheter et de vendre des

produits et services, tant licites qu'illicites, tout en préservant leur anonymat grâce à l'utilisation de Tor et des paiements en bitcoins. Silk Road proposait une gamme variée d'articles, principalement des drogues et des médicaments normalement sur ordonnance, mais excluait des activités jugées contraires aux principes jusnaturalistes, telles que la vente de services de tueurs à gages ou de contenus pédopornographiques.

La plateforme reflétait les convictions agoristes de Ross Ulbricht, en promouvant un marché libre et non régulé, échappant ainsi au contrôle des gouvernements et à la taxation. Silk Road a joué un rôle significatif dans l'adoption initiale de Bitcoin au début des années 2010, en contribuant à sa notoriété et à son utilisation au-delà des cercles *cypherpunks*.

Silk Road a été fermé en 2013. Ross Ulbricht a été arrêté la même année et condamné à deux peines de réclusion à perpétuité plus 40 ans, sans possibilité de libération conditionnelle, assortie d'une restitution de 183 millions de dollars. Cette condamnation soulève des questions sur la proportionnalité des peines attribuées pour des crimes en ligne non violents. Ross Ulbricht a finalement été gracié par le président Donald Trump le 21 janvier 2025, après avoir passé plus de 11 ans en prison.

Termes associés : **AGORISME** | **TOR**

SIMPLICITY

SCRIPT

Langage de programmation bas niveau conçu par Russell O'Connor chez Blockstream pour exprimer des contrats sur des blockchains qui reposent sur le modèle d'UTXO. Présenté dans un livre blanc en 2017, Simplicity se distingue des langages comme Solidity (Ethereum) par le fait qu'il n'est pas Turing-complet : il peut exprimer toute fonction finie, mais interdit les boucles non bornées et l'état global. Cette restriction permet de garantir que chaque exécution se termine, et que le coût en ressources d'un programme est connu avant son exécution. Le langage a été déployé sur Liquid en juillet 2025, où il offre aux émetteurs d'actifs la possibilité de créer des contrats vérifiables. Contrairement à Miniscript, qui se limite aux signatures, timelocks et préimages de hachage au sein de Script, Simplicity peut exprimer des fonctions arbitraires, mais nécessite un environnement de consensus distinct de celui de Bitcoin.

Termes associés : **LIQUID NETWORK** | **MINISCRIPT**

SIMPLIFIED PAYMENT VERIFICATION

PROTOCOLE | FR : VÉRIFICATION SIMPLIFIÉE DE PAIEMENT

Méthode permettant aux clients légers de vérifier les transactions Bitcoin sans télécharger l'intégralité de la blockchain. Un nœud qui utilise SPV télécharge uniquement les entêtes de blocs qui sont beaucoup plus légers que les blocs complets. Lorsqu'il doit vérifier une transaction, le nœud SPV demande une preuve de Merkle aux nœuds complets pour confirmer que la transaction est incluse dans un bloc spécifique. Cette approche est efficace pour les appareils avec des ressources limitées, comme les smartphones, mais implique une dépendance vis-à-vis des nœuds complets, ce qui peut réduire la sécurité et augmenter la confiance requise.

On utilise souvent le sigle « SPV » pour évoquer cette méthode.

Terme associé : **NOEUD SPV - NOEUD LÉGER**

SINGLE-SIG

PORTEFEUILLE

Configuration de portefeuille Bitcoin dans laquelle une seule clé privée suffit pour signer une transaction et autoriser la dépense des fonds. Ce mode de fonctionnement, aussi appelé « signature unique », constitue la forme la plus simple et la plus répandue de sécurisation d'un portefeuille. Contrairement au multisig, qui exige plusieurs signatures qui proviennent de différentes clés pour valider une transaction, le single-sig ne repose que sur la détention d'une seule seed.

Termes associés : **CLÉ PRIVÉE** | **MULTISIG** | **SIGNATURE NUMÉRIQUE**

SINGLE-USE SEAL

RGB

Désigne une promesse d'engagement cryptographique relative à un message encore inconnu, lequel sera révélé une seule fois à l'avenir et devra être accessible à l'ensemble d'une audience déterminée. Ce mécanisme vise à prévenir la création de multiples engagements concurrents pour un même sceau, ce qui garantit l'unicité de chaque validation. En imposant cette contrainte d'utilisation unique, le Single-use Seal est utilisé sur RGB pour éviter la double-dépense.

Termes associés : **CLIENT-SIDE VALIDATION** | **RGB**

SLIP

COMMUNAUTÉ

Sigle de « *SatoshiLabs Improvement Proposals* ». C'est un ensemble de propositions visant à améliorer ou à standardiser l'utilisation de Bitcoin, émanant de SatoshiLabs, la société à l'origine des portefeuilles matériels Trezor. Ces propositions s'articulent souvent comme des extensions de BIP (*Bitcoin Improvement Proposals*), dans le but d'enrichir les standards existants. Elles exposent les décisions techniques prises par SatoshiLabs qui ne trouvent pas leur place dans les BIP, mais qui restent pertinentes pour d'autres développeurs de logiciels de portefeuilles ou de portefeuilles matériels, notamment pour contribuer à l'uniformisation des processus.

Termes associés : **BIP** **HARDWARE WALLET**

SMALL BLOCKERS

HISTOIRE

Nom donné aux partisans du maintien de la taille des blocs de Bitcoin à un niveau bas durant la Blocksize War entre 2015 et 2017, afin de préserver la décentralisation et la sécurité du système. Ils soutiennent des solutions comme SegWit, pensent que le passage à l'échelle doit se faire sur des couches supérieures comme le Lightning Network, et soutiennent la mise en place des évolutions via des soft forks plutôt que des hard forks.

Terme associé : **BLOCKSIZE WAR**

SMPPS

MINAGE

Sigle de « *Shared Maximum Pay Per Share* ». C'est une méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage. C'est une variante de la méthode PPS. Elle limite les paiements de sorte que la pool ne paie jamais plus que ce qu'elle a gagné en récompenses de blocs. Ainsi, même si les mineurs soumettent des shares valides, la récompense totale distribuée ne peut excéder les revenus de la pool. Cette méthode vise à maintenir l'équilibre financier de la pool tout en utilisant un système de paiement à la tâche, qui permet de réduire la variance des mineurs individuels.

Terme associé : **SHARES**

SMT - SPARSE MERKLE TREE

</> INFORMATIQUE

Structure de données qui étend les propriétés d'un arbre de Merkle classique en permettant de produire des preuves de non-inclusion en plus des preuves d'inclusion. Dans un SMT, chaque élément est placé à une position dans l'arbre déterminée par le hash SHA-256 de ses données : chaque bit du hash indique un parcours gauche ou droite dans l'arbre binaire. La majorité des feuilles restant vides, l'arbre est dit « creux » (*sparse*), ce qui permet une génération et une mise à jour efficaces grâce à l'élagage des branches vides. Les SMT sont utilisés dans le protocole Taproot Assets pour gérer les transferts d'actifs et prouver qu'un actif donné n'a pas été dépensé deux fois.

Termes associés : **ARBRE DE MERKLE** | **TAPROOT ASSETS PROTOCOL**

SNAPSHOT

⚙ PROTOCOLE

Capture instantanée de l'état de l'UTXO set de Bitcoin à un bloc donné. Un snapshot enregistre l'intégralité des UTXOs existants à cette hauteur de bloc, ce qui forme une photographie complète de toutes les pièces non dépensées sur le réseau.

L'utilisation principale du snapshot dans Bitcoin se fait au travers du mécanisme Assume UTXO, intégré dans le client Bitcoin Core. Lors de l'initialisation d'un nouveau nœud, ce dernier peut charger un snapshot de l'UTXO set qui est précalculé et dont l'empreinte cryptographique est vérifiée. Cette approche permet au nœud d'être fonctionnel presque immédiatement, sans avoir à valider l'intégralité de l'historique de la blockchain depuis le bloc de genèse. La vérification complète de la chaîne est ensuite réalisée en arrière-plan.

Termes associés : **ASSUME UTXO** | **NOEUD COMPLET** | **UTXO SET**

SNARK

🔒 CRYPTOGRAPHIE

Système de preuve cryptographique dont l'acronyme signifie *Succinct Non-interactive ARgument of Knowledge*. Un SNARK permet à un prouveur de convaincre un vérificateur qu'une affirmation est vraie, sans interaction et sans révéler les données sous-jacentes, en produisant une preuve de taille très réduite (quelques centaines d'octets) vérifiable en temps constant. Cette concision est l'avantage principal des SNARK par rapport aux STARK. En revanche, la plupart des constructions SNARK, comme Groth16, nécessitent une cérémonie de configuration de confiance (*trusted setup*) spécifique au circuit à prouver. Si les paramètres secrets de cette cérémonie sont compromis, il devient possible de forger de fausses preuves. Les SNARK reposent sur des hypothèses de difficulté liées aux courbes elliptiques et ne sont donc pas intrinsèquement résistants aux attaques quantiques. Dans l'écosystème Bitcoin, les SNARK sont utilisés notamment pour compresser des preuves STARK plus volumineuses avant leur publication on-chain, comme dans le cas du rollup Citrea.

Termes associés : **GROTH16** | **STARK**

SO - SPARK OPERATOR

COUCHE SUPÉRIEURE

Nœud individuel au sein de la *Spark Entity* (SE), participant au protocole de signature à seuil qui sécurise les transferts *off-chain* dans Spark. Chaque SO détient un fragment de la part de clé de la SE, sans jamais posséder la clé complète à lui seul. L'ensemble des SO (ou un seuil d'entre eux) coopèrent via le protocole FROST pour co-signer les transactions et renouveler les parts de clé lors des transferts de propriété.

Les SO jouent également un rôle de *watchtower* : ils surveillent la blockchain Bitcoin pour détecter d'éventuelles tentatives d'exit frauduleuses par d'anciens propriétaires.

Si un SO est indisponible, les transferts peuvent être temporairement bloqués, mais les utilisateurs conservent la possibilité de sortir unilatéralement vers la couche principale. Au lancement de Spark, deux opérateurs (Lightspark et Flashnet) composent la SE, avec l'objectif d'élargir le nombre d'opérateurs à des juridictions diverses pour diffuser le risque.

Termes associés : **SE - SPARK ENTITY** | **SPARK**

SOFT FORK

PROTOCOLE

Modification des règles du protocole de manière rétrocompatible. Contrairement au hard fork, le soft fork ne donne pas lieu à une séparation du réseau de nœuds Bitcoin en deux groupes distincts, à condition qu'une majorité de la puissance de calcul se trouve sur la chaîne à jour. Si tout se passe bien, les nœuds avec la mise à jour et les nœuds sans la mise à jour restent donc sur la même blockchain. Une modification est rétrocompatible lorsqu'elle ajoute ou rend plus restrictives certaines règles du protocole.

Termes associés : **HARD FORK** | **UASF**

SOLO MINING

MINAGE | FR : MINAGE EN SOLO

Pratique de minage dans laquelle un mineur utilise ses propres équipements pour chercher des preuves de travail valides de manière autonome, sans rejoindre une pool de minage. Le mineur en solo exploite son propre nœud complet pour récupérer les transactions en attente, construire ses blocs candidats et tenter de trouver un hash satisfaisant la cible de difficulté du réseau.

En *solo mining*, le mineur conserve l'intégralité de la récompense de bloc lorsqu'il parvient à en trouver un valide, sans frais de pool à payer, et garde un contrôle total sur la sélection des transactions incluses dans ses blocs. En revanche, la probabilité de trouver un bloc est directement proportionnelle au hashrate du mineur par rapport au hashrate total du réseau. Pour un mineur individuel disposant d'un hashrate modeste, cette probabilité est extrêmement faible, ce qui rend les revenus très irréguliers et imprévisibles. C'est cette forte variance qui a motivé l'émergence des pools de minage, où les mineurs mutualisent leur puissance de calcul pour obtenir des rémunérations plus fréquentes

et régulières.

Termes associés : **MINAGE** | **MINING POOL**

SOMME DE CONTRÔLE

CRYPTOGRAPHIE | EN : CHECKSUM

La somme de contrôle est une valeur calculée à partir d'un ensemble de données, utilisée pour vérifier l'intégrité et la validité de ces données lors de leur transmission ou de leur stockage. Les algorithmes de somme de contrôle sont conçus pour détecter des erreurs accidentelles ou des altérations involontaires des données, comme les erreurs de transmission ou les corruptions de fichiers. Différents types d'algorithmes de somme de contrôle existent, tels que le contrôle de parité, les sommes de contrôle modulaires, les fonctions de hachage cryptographiques, ou encore les codes BCH (*Bose, Ray-Chaudhuri et Hocquenghem*).

Sur Bitcoin, les sommes de contrôle sont utilisées au niveau applicatif pour assurer l'intégrité des adresses de réception. Une somme de contrôle est calculée à partir de la charge utile d'une adresse d'un utilisateur, puis ajoutée à cette adresse afin de détecter d'éventuelles erreurs lors de sa saisie. Une somme de contrôle est également présente dans les phrases de récupération (mnémonique).

Il est généralement admis d'utiliser directement le terme anglais de « checksum » en français.

Termes associés : **ADRESSE DE RÉCEPTION** | **BCH CODE**

SOROBAN

CONFIDENTIALITÉ

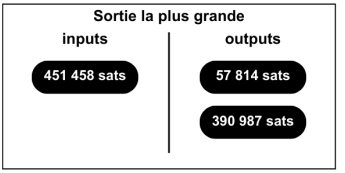
Protocole de communication chiffré établi sur Tor permettant de collaborer avec d'autres utilisateurs dans le cadre d'une transaction Cahoots. Soroban a été développé par les équipes de Samourai Wallet, afin de faciliter l'échange de transactions partiellement signées (PSBT) entre les utilisateurs qui souhaitent réaliser des transactions collaboratives (*Stowaway*, *Stonewall x2*...). Ce protocole est utilisé sur l'application Samourai Wallet et sur le logiciel Sparrow Wallet.

Termes associés : **CAHOOTS** | **PSBT**

SORTIE LA PLUS GRANDE

CONFIDENTIALITÉ | EN : LARGEST OUTPUT

Heuristique d'analyse de chaîne interne. Lorsque l'on repère un écart suffisamment large entre 2 sorties de transaction sur un modèle de paiement simple, on peut estimer que la sortie la plus grande est vraisemblablement le change. Cette heuristique du plus gros output est sûrement la plus imprécise de toutes. Si on l'identifie seule, elle est assez faible. Toutefois, cette caractéristique peut être additionnée avec d'autres heuristiques, afin de réduire l'incertitude de notre interprétation.



Termes associés : **ANALYSE DE CHAINE** | **CHANGE**

SORTIE NON RENTABLE

ATTAQUE | EN : UNECONOMICAL OUTPUT

Fait référence à des montants de pièces bitcoin extrêmement petits qui sont trop minimes pour être envoyés dans une transaction, car les frais nécessaires pour les inclure dans un bloc seraient proportionnellement plus élevés que leur valeur. La définition précise de « dust » peut varier selon le contexte, mais il s'agit généralement de toute sortie de transaction qui nécessite plus de frais pour être dépensée qu'elle n'incarne de valeur. Pour l'utilisateur de Bitcoin, il est important de gérer ses UTXOs, et de pratiquer la consolidation de ceux-ci, afin qu'ils ne deviennent pas du dust.

En anglais, on croise parfois ce terme de « uneconomical outputs » pour désigner du dust.

Termes associés : **DUST** | **UTXO**

SOURCEFORGE

INFORMATIQUE

Plateforme web d'hébergement et de distribution de logiciels *open source*, fondée en 1999. SourceForge a joué un rôle historique dans les débuts de Bitcoin : le projet y a été enregistré par Satoshi Nakamoto en novembre 2008, et la première version du logiciel (v0.1) y a été publiée en janvier 2009. La plateforme a également hébergé le forum de discussion initial de la communauté Bitcoin, ainsi que la liste de diffusion bitcoin-dev à ses débuts. Le projet Bitcoin a ensuite migré vers GitHub, devenu le principal outil de collaboration pour le développement du protocole.

Terme associé : **GITHUB**

SPARK

COUCHE SUPÉRIEURE

Protocole de couche supérieure open source construit sur Bitcoin, développé par Lightspark, conçu pour permettre des transactions instantanées et à très faible coût tout en préservant la self-custody des utilisateurs. Spark repose principalement sur la technologie des statechains : plutôt que de déplacer les bitcoins on-chain à chaque transfert, le protocole modifie l'autorisation de dépense partagée entre l'utilisateur et un groupe d'opérateurs appelé *Spark Entity* (SE). La SE est composée de plusieurs *Spark Operators* (SO) qui participent conjointement à un protocole de signature partagée. Il suffit qu'un seul opérateur soit honnête pour empêcher toute collusion avec un ancien propriétaire.

Contrairement au Lightning Network qui nécessite la gestion de canaux de paiement, Spark transfère la propriété d'UTXO hors chaîne sans contrainte de liquidité ni de gestion de canaux. Le protocole est nativement compatible avec Lightning : les utilisateurs peuvent envoyer et recevoir des paiements Lightning sans opérer de nœud. Spark supporte également l'émission de jetons (BTKN) et de stablecoins directement sur Bitcoin. L'utilisateur conserve à tout moment la possibilité de retirer ses fonds vers la couche principale de manière unilatérale, sans permission de la SE. Les transferts entre portefeuilles Spark se règlent en moins d'une seconde, sans frais.

Termes associés : **LIGHTNING NETWORK** | **STATECHAIN**

SPARK SERVICE PROVIDER

COUCHE SUPÉRIEURE

Fournisseur de services au sein de l'écosystème Spark, dont le rôle principal est de faciliter les dépôts et retraits entre la couche principale de Bitcoin et Spark. Plusieurs SSP peuvent coexister au sein d'un même réseau Spark.

Les SSP assurent également l'interopérabilité avec le Lightning Network en agissant comme intermédiaires pour les paiements Lightning. Lorsqu'un utilisateur Spark souhaite payer une facture Lightning, le SSP exécute un *atomic swap*.

Termes associés : **LIGHTNING SERVICE PROVIDER** | **SPARK**

SPARK TREE

COUCHE SUPÉRIEURE | FR : ARBRE SPARK

Structure arborescente offchain utilisée dans le protocole Spark pour étendre un UTXO unique en plusieurs sous-unités de valeur, afin de résoudre le problème des statechains classiques qui imposaient le transfert d'un UTXO entier. L'arbre se compose de branches (*branch transactions*) et de feuilles (*leaf transactions*).

La racine de l'arbre correspond à l'UTXO on-chain (`Txn0`), dont la clé de l'opérateur est définitivement détruite. Les branches sont des transactions intermédiaires maintenues hors chaîne, sans timelock, qui ne peuvent être dépensées que par l'agrégation des clés de toutes les feuilles situées en dessous. Les feuilles sont les transactions terminales détenues par les utilisateurs individuels, chacune associée à une exit transaction avec un timelock relatif.

La division d'une feuille repose sur une propriété d'additivité des clés : la clé parente est scindée en plusieurs clés enfants dont la somme est égale à la clé originale ($a_0 = a_1 + a_2 + \ldots + a_n$). Cette propriété permet aussi la ré-agrégation des feuilles.

Termes associés : **EXIT TRANSACTION** | **SPARK**

SPARKSCAN

✱ OUTIL

Explorateur et suite d'analyse publique dédiée au protocole Spark. Sparkscan permet de consulter les adresses, les transactions, les jetons (BTKN) et l'activité en temps réel du réseau Spark.

Termes associés : **EXPLORATEUR DE BLOC** | **SPARK**

SPARROW WALLET

⊸ PORTEFEUILLE

Logiciel de portefeuille Bitcoin pour ordinateur (Windows, macOS, Linux), conçu pour les utilisateurs qui souhaitent un contrôle complet sur leurs transactions Bitcoin. Sparrow prend en charge la plupart des *hardware wallets* du marché via PSBT, le multisig, le *coin control*, le *labelling*... Il permet la connexion à son propre nœud Bitcoin (Bitcoin Core ou serveur Electrum) pour une vérification souveraine des transactions, mais peut aussi fonctionner avec un serveur public. Le logiciel est open source.

Termes associés : **COINJOIN** | **HARDWARE WALLET** | **PSBT**

SPECTER

⊸ PORTEFEUILLE

Suite logicielle open source dédiée à la gestion de portefeuilles Bitcoin, avec un accent particulier sur les configurations multisig. Specter Desktop est une application de bureau qui s'interface directement avec un nœud Bitcoin Core personnel pour créer et gérer des portefeuilles singlesig et multisig, compatible avec la plupart des *hardware wallets* du marché (Trezor, Ledger, Coldcard, etc.). Le projet inclut également Specter DIY, un *firmware* open source qui permet de construire son propre *signing device* à partir de composants électroniques standards.

Termes associés : **BITCOIN CORE** | **MULTISIG**

SPEEDY TRIAL

PROTOCOLE

Méthode d'activation de soft fork initialement conceptualisée pour Taproot début 2021 par David A. Harding sur une idée de Russell O'Connor. Son principe est d'utiliser la méthode du BIP-0008 avec un paramètre `LOT` réglé sur `faux`, tout en réduisant le délai d'activation à seulement 3 mois. Cette réduction du délai de vote permet une vérification rapide de l'approbation des mineurs. Si le seuil d'approbation requis est atteint pendant l'une des périodes, le soft fork est alors verrouillé. Il sera activé plusieurs mois plus tard, donnant ainsi aux mineurs le temps nécessaire pour mettre à jour leurs logiciels.

Le succès de cette méthode pour Taproot, qui bénéficiait d'un large consensus au sein de la communauté Bitcoin, ne garantit cependant pas son efficacité pour toutes les mises à jour. Bien que la méthode Speedy Trial permette une activation plus rapide, certains développeurs expriment des inquiétudes quant à son utilisation future. Ils craignent qu'elle ne conduise à une succession trop rapide de soft forks, ce qui pourrait potentiellement menacer la stabilité et la sécurité du protocole Bitcoin. Par rapport au BIP-0008 avec le paramètre `LOT=true`, la méthode Speedy Trial est beaucoup moins menaçante pour les mineurs. Aucun UASF n'est prévu automatiquement. Cette méthode d'activation n'a pas encore été formalisée au sein d'un BIP.

« Speedy Trial » est emprunté d'une terminologie juridique qui indique un « procès expéditif ». Cela invoque le fait que la proposition d'amélioration est envoyée rapidement devant le tribunal des mineurs, afin que l'on soit fixés sur leurs intentions. Il est généralement admis d'utiliser directement le terme anglais en français.

Termes associés : **BIP-0008** **SOFT FORK**

SPHINX

LIGHTNING NETWORK

Format de paquets cryptographiques utilisé par le Lightning Network pour acheminer les paiements de manière privée via le routage en oignon. Conçu à l'origine par George Danezis et Ian Goldberg en 2009 pour les réseaux de mixage, il encapsule les instructions de routage dans des couches de chiffrement successives. Chaque nœud intermédiaire ne peut déchiffrer que la couche qui lui est destinée, révélant uniquement le prochain saut sans connaître l'expéditeur ni le destinataire final.

Termes associés : **HTLC** **ROUTAGE EN OIGNON**

SPIDERPOOL

MINAGE

Pool de minage de Bitcoin fondée en Chine. SpiderPool compte parmi les principales pools de minage au monde en termes de hashrate. Elle propose des méthodes de rémunération FPPS et PPLNS, et s'adresse aussi bien aux mineurs individuels qu'aux fermes de minage professionnelles.

Terme associé : **MINING POOL**

SPIRAL

ORGANISATION

Filiale de Block Inc. (anciennement Square) dédiée au financement et au développement de projets Bitcoin *open source*. Créée en 2019 sous le nom de « Square Crypto », puis renommée « Spiral » en 2021, l'organisation finance des développeurs indépendants, des designers et des contributeurs qui travaillent sur Bitcoin et son écosystème.

Spiral a notamment soutenu des projets comme Bitcoin Core ou BTCPay Server. L'organisation développe aussi le Lightning Dev Kit (LDK) et le Bitcoin Dev Kit (BDK), deux bibliothèques *open source* qui facilitent l'intégration de Bitcoin et Lightning dans des applications tierces.

Terme associé : **BITCOIN CORE**

SPLICING

LIGHTNING NETWORK | FR : RACCORDEMENT DE CANAL

Mécanisme du Lightning Network qui permet de redimensionner un canal de paiement sans le fermer, soit en y ajoutant des fonds (*splice-in*), soit en retirant des fonds vers une sortie on-chain (*splice-out*). Le canal reste pleinement opérationnel pendant toute l'opération.

Le splicing crée une nouvelle transaction multisignature 2-de-2 qui dépense l'UTXO de financement existant vers un nouvel UTXO avec la capacité ajustée. Le canal entre d'abord en mode de « *quiescence* » (STFU) pour geler son état, puis les deux parties négocient la transaction via le protocole interactive-tx, où chacune peut ajouter des entrées et des sorties. Des tiers peuvent également joindre leurs propres opérations à cette même transaction, ce qui permet de partager les frais on-chain. Pendant la confirmation, l'ancien et le nouvel état du canal coexistent, ce qui permet de continuer à acheminer des paiements sans interruption.

Le splicing réduit de deux à une seule le nombre de transactions on-chain nécessaires pour ajuster la capacité d'un canal. Il rend aussi possible les portefeuilles à solde unique qui unifient les fonds on-chain et Lightning en une seule balance pour l'utilisateur.

Termes associés : **CANAL DE PAIMENT** | **LIGHTNING NETWORK**

SPOF

‹/› INFORMATIQUE | FR : POINT DE DÉFAILLANCE UNIQUE

Un point de défaillance unique (SPOF, de l'anglais « *Single Point of Failure* ») désigne dans le domaine informatique un composant ou un élément d'un système dont la défaillance entraînerait l'arrêt complet ou une perte significative de fonctionnalités de l'ensemble du système. Il peut s'agir d'une pièce matérielle, d'une information, d'un logiciel, ou d'une partie d'un réseau. Par exemple, dans le contexte spécifique des portefeuilles HD Bitcoin, la phrase de récupération de 12 ou de 24 mots constitue souvent un SPOF pour le portefeuille. Si son secret n'est pas assuré, l'intégralité du portefeuille pourrait être subtilisée. De la même manière, sa simple perte pourrait entraîner la perte de l'intégralité des bitcoins du portefeuille.

Termes associés : **PHRASE DE RÉCUPÉRATION** | **PORTEFEUILLE**

SPREAD - WST

CONFIDENTIALITÉ | FR : DIFFUSION - WST

Dans le logiciel Whirlpool Stat Tool, le spread est un indicateur permettant de mesurer l'homogénéité du processus de mixage du point de vue d'une pièce donnée. On différenciera 2 spreads : le prospectif et le rétrospectif.

Le spread prospectif est calculé en tant que ratio entre l'anonset prospectif de votre pièce et le nombre total de pièces créées après votre Tx0. Par exemple, si dans votre pool il y a 100 pièces et que votre pièce a un anonset de 70, le spread prospectif de votre pièce est alors de 70 %.

Le spread rétrospectif, quant à lui, est le ratio entre l'anonset rétrospectif de votre pièce et le nombre total de Tx0 créées avant le dernier mix de votre pièce. Ainsi, si l'anonset rétrospectif de votre pièce est de 95 et qu'il y a eu 100 Tx0 avant votre dernier mix, alors le spread rétrospectif de votre pièce est de 95 %.

Ces deux indicateurs permettent d'évaluer l'efficacité du mixage de votre pièce par rapport au potentiel offert par la pool. Un spread prospectif faible, comme 5 % par exemple, indique une importante marge d'amélioration possible par des mixages supplémentaires. Inversement, un spread prospectif élevé, comme par exemple 97 %, signifie que peu d'anonset supplémentaire peut être gagné avec des coinjoins supplémentaires.

Termes associés : **ANONSETS - ANONYMITY SETS** | **WHIRLPOOL STAT TOOL**

SPY MINING

↗ MINAGE | FR : MINAGE ESPION

Pratique dans laquelle un mineur ou une pool de minage se connecte à une pool concurrente afin de recevoir le hash de l'en-tête de bloc dès qu'un nouveau bloc est découvert par cette pool. En obtenant cette information avant sa diffusion sur le réseau, le mineur espion peut immédiatement commencer à miner sur le nouveau bloc, sans attendre sa réception et sa validation complète par le réseau.

Cette technique procure un avantage de quelques secondes par rapport aux mineurs qui attendent la propagation normale du bloc. Le *spy mining* est une pratique courante dans l'industrie du minage, qui contribue à un environnement concurrentiel où les pools sont en compétition non seulement avec le reste du réseau, mais aussi entre elles. Il est souvent associé au *validationless mining*, car le mineur espion commence à travailler sur un nouveau bloc sans en avoir vérifié l'intégralité des transactions.

Termes associés : **MINING POOL** | **SELFISH MINING**

SRC-20

COUCHE SUPÉRIEURE

Norme de token fongible sur Bitcoin qui repose sur le protocole STAMPS pour inscrire les métadonnées directement dans les UTXO. Contrairement à la norme BRC-20, qui stocke ses données dans la partie *witness* des transactions (susceptible d'être élaguée par les nœuds), les données SRC-20 sont inscrites dans les sorties non dépensées, ce qui les rend permanentes et non supprimables.

Termes associés : **BRC-20** | **STAMPS**

STABLECOIN

ÉCONOMIE ET RÉGULATION

Catégorie de cryptomonnaie conçue pour maintenir une valeur stable par rapport à un actif référence, souvent une monnaie étatique comme le dollar américain.

Termes associés : **ALTCOIN** | **CRYPTOMONNAIE**

STACKER

COMMUNAUTÉ

Terme utilisé dans la communauté Bitcoin pour désigner l'action d'accumuler régulièrement des sats. C'est une stratégie d'épargne à long terme qui consiste à acheter du bitcoin de manière récurrente, indépendamment du prix du marché, souvent via la méthode du DCA (*Dollar-Cost Averaging*).

Termes associés : **HODL** | **SAT - SATOSHI**

STALE SHARE

↗ MINAGE | FR : PART PÉRIMÉE

Dans le cadre d'une pool de minage, une stale share est une share (part de travail) soumise par un mineur après qu'un nouveau bloc a déjà été trouvé, rendant le travail effectué sur l'ancien bloc obsolète. Cela se produit généralement en raison de la latence réseau entre le mineur et le serveur de la pool : le mineur continue de calculer des hashs sur un bloc qui n'est plus d'actualité, car il n'a pas encore reçu la notification du nouveau bloc.

Les stale shares sont rejetées par la pool et ne sont pas comptabilisées dans le calcul de la rémunération du mineur. Un taux de stale shares autour de 1 % est considéré comme normal. Un taux plus élevé peut indiquer des problèmes de connexion réseau, une latence excessive entre le mineur et le serveur de la pool, ou un hashrate très élevé qui génère des shares plus rapidement que le réseau ne peut les transmettre. Pour réduire le taux de stale shares, les mineurs peuvent privilégier une connexion Ethernet plutôt que Wi-Fi, choisir une pool géographiquement proche de leurs équipements, ou utiliser un proxy de minage.

Termes associés : **MINING POOL** | **SHARES**

STAMPS

COUCHE SUPÉRIEURE

Protocole qui permet d'intégrer des données d'image formatées directement sur la blockchain Bitcoin via des transactions multisignatures brutes (P2MS). Stamps encode le contenu binaire d'une image en base64 et l'ajoute dans les clés d'un P2MS 1/3. Une clé est réelle et sert à dépenser les fonds, tandis que les deux autres sont des fausses clés (on ne connait pas la clé privée associée) qui stockent les données. En encodant les données directement sous forme de clés publiques plutôt qu'en utilisant des sorties `OP_RETURN`, les images stockées avec le protocole Stamps sont particulièrement intensives en termes de charge de travail pour les nœuds. Cette méthode crée notamment de multiples UTXOs, ce qui augmente la taille de l'UTXO set et pose des problèmes pour les nœuds complets.

Termes associés : **UTREEXO** | **UTXO SET**

STARK

CRYPTOGRAPHIE

Système de preuve cryptographique dont l'acronyme signifie *Scalable Transparent ARgument of Knowledge*. Les STARK permettent à un prouveur de démontrer la validité d'un calcul à un vérificateur, sans révéler les données sous-jacentes et sans nécessiter de configuration de confiance initiale (*trusted setup*). Cette propriété de « transparence » les distingue des SNARK, qui requièrent généralement une cérémonie de configuration préalable. Les STARK reposent sur des fonctions de hachage et sur le protocole FRI (*Fast Reed-Solomon Interactive Oracle Proof of Proximity*), ce qui les rend résistants aux attaques quantiques. En contrepartie, les preuves STARK sont significativement plus volumineuses que les preuves SNARK (de l'ordre de plusieurs dizaines de kilo-octets), ce qui les rend moins adaptées à la vérification directe on-chain. Pour contourner cette limitation, il est courant d'encapsuler une preuve STARK dans une preuve SNARK plus compacte avant publication. C'est l'approche adoptée par le rollup Citrea sur Bitcoin, via le framework RISC Zero.

Termes associés : **RISC ZERO** | **SNARK**

START9

OUTIL

Entreprise américaine (Start9 Labs, Inc.) développant StartOS, un système d'exploitation libre pour serveurs personnels. À l'instar d'Umbrel, Start9 vise à démocratiser l'auto-hébergement en permettant à des utilisateurs non techniques d'installer et de gérer des applications *open source* sur leur propre matériel, sans dépendre de services *cloud* tiers.

Start9 commercialise des serveurs préconfigurés et propose un *marketplace* d'applications auto-hébergées avec notamment Bitcoin Core, des implémentations de Lightning, des solutions de stockage et de communication, etc.

Termes associés : **LIGHTNING NETWORK** | **UMBREL**

STASH

RGB

Dans le cadre du protocole RGB, le Stash désigne l'ensemble des données stockées côté client par un utilisateur pour un ou plusieurs contrats. Ces données comprennent notamment l'historique des transitions, les *consignments* ainsi que les preuves de validité, indispensables au processus de validation côté client (*Client-side Validation*). Chaque détenteur conserve uniquement les fragments pertinents de l'historique (*shards*) nécessaires à la vérification et à la mise à jour de ses contrats.

Termes associés : **CLIENT-SIDE VALIDATION** | **CONSIGNMENT**

STATE EXTENSION

RGB

Dans le cadre du protocole RGB, une *State Extension* désigne une opération contractuelle permettant d'initier une mise à jour de l'état d'un contrat via la rédemption de *Valencies* préalablement déclarées. Cette procédure exploite un droit public inscrit dans le contrat pour déclencher une modification, qui ne devient effective qu'après avoir été clôturée par une *State Transition*. Cette dernière intègre la mise à jour définitive de l'état du contrat.

Termes associés : **STATE TRANSITION** | **VALENCY**

STATE TRANSITION

RGB

Dans le cadre du protocole RGB, la *State Transition* désigne l'opération permettant de faire évoluer l'état d'un contrat vers une nouvelle configuration. Elle peut modifier tant les données du *Global State* que celles des *Owned States*. Chaque transition est strictement vérifiée selon les règles définies dans le *Schema* du contrat, ce qui garantit la conformité des modifications aux contraintes établies lors de la *Genesis*. Cette opération est ancrée dans la blockchain Bitcoin via un *commitment*.

Termes associés : **GENESIS** | **SCHEMA**

STATECHAIN

COUCHE SUPÉRIEURE

Mécanisme de couche supérieure sur Bitcoin permettant de transférer la propriété d'un UTXO hors chaîne, sans diffuser de transaction on-chain. Le concept a été proposé initialement par Ruben Somsen en 2018. Il repose sur un partage de clé entre l'utilisateur et une entité (*Statechain Entity*). L'UTXO reste verrouillé on-chain dans une adresse multisignature, mais le contrôle effectif change de mains par un renouvellement des parts de clé privée : lorsque Alice transfère à Bob, l'entité détruit sa part associée à Alice et en génère une nouvelle avec Bob. L'ancienne part d'Alice devient inutilisable.

La sécurité repose sur l'hypothèse que l'entité supprime effectivement ses anciennes parts de clé. Pour réduire ce risque de confiance, l'entité peut être distribuée entre plusieurs opérateurs via un protocole de signature à seuil : il suffit qu'un seul opérateur soit honnête pour garantir l'intégrité du transfert. Un mécanisme de sortie unilatérale permet à l'utilisateur de récupérer ses fonds on-chain. Les statechains sont utilisées notamment par le protocole Spark pour les transferts instantanés de bitcoins et d'autres tokens.

Termes associés : **SPARK** | **UTXO**

STEALTH ADDRESS

CONFIDENTIALITÉ | FR : ADRESSE FURTIVE

Technique de confidentialité qui permet à un destinataire de recevoir des paiements sur des adresses à usage unique, dérivées à la volée par l'expéditeur, sans que ces adresses puissent être publiquement reliées à l'identité du destinataire. Le principe repose sur la cryptographie sur courbes elliptiques : le destinataire publie une clé publique statique (la *stealth address*), et chaque expéditeur génère une adresse de réception éphémère en combinant cette clé publique avec un secret partagé dérivé par ECDH (*Elliptic Curve Diffie-Hellman*). Seul le destinataire, grâce à sa clé privée, peut identifier et dépenser les fonds reçus sur ces adresses éphémères. Cette approche évite la réutilisation d'adresses. Le concept de *stealth address* a été initialement proposé par un utilisateur du forum BitcoinTalk sous le pseudonyme « ByteCoin » en 2011, puis formalisé par Peter Todd en janvier 2014, et a servi de fondement pour la création des protocoles BIP-0047 et *Silent Payments*.

Termes associés : **SIGNATURE NUMÉRIQUE** | **UTXO**

STÉGANOGRAPHIE

CRYPTOGRAPHIE | EN : STEGANOGRAPHY

La stéganographie est une technique consistant à dissimuler une information sensible dans un objet ou média de couverture d'apparence anodine (typiquement un signal de type texte, son, image, vidéo, transaction Bitcoin, etc.), qui n'éveillera pas les soupçons même si elle est à la vue de tous ; seul un destinataire légitime saura qu'il y a là une information pour lui et où la chercher.

Termes associés : **CHIFFRER - CHIFFREMENT** | **CRYPTANALYSE**

STONEWALL

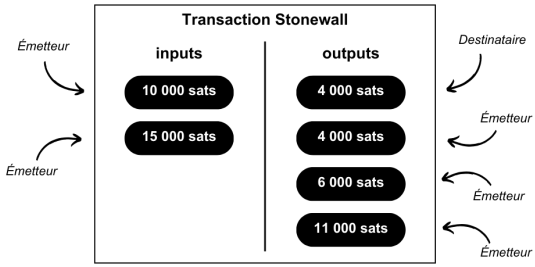
CONFIDENTIALITÉ

Forme spécifique de transaction Bitcoin visant à accroître la confidentialité des utilisateurs lors d'une dépense en imitant un coinjoin entre deux personnes, sans pour autant en être un. En effet, cette transaction n'est pas collaborative. Un utilisateur peut la construire tout seul, en faisant uniquement intervenir les UTXOs lui appartenant en inputs. Vous pouvez donc créer une transaction Stonewall pour n'importe quelle occasion, sans avoir besoin de vous synchroniser avec un autre utilisateur.

Le fonctionnement de la transaction Stonewall est le suivant : en entrée de la transaction, l'émetteur utilise 2 UTXOs qui lui appartiennent. En sortie, la transaction produit 4 outputs, dont 2 qui seront exactement de même montant. Les 2 autres constitueront du change. Parmi les 2 outputs de même montant, un seul ira effectivement au destinataire du paiement.

Il y a donc seulement 2 rôles dans une transaction Stonewall :

- L'émetteur, qui réalise le paiement effectif ;
- Le destinataire, qui peut ignorer la nature spécifique de la transaction et attend simplement un paiement de la part de l'émetteur.



Les transactions Stonewall sont disponibles à la fois sur l'application Samourai Wallet et sur le logiciel Sparrow Wallet.

La structure Stonewall ajoute énormément d'entropie à la transaction et vient brouiller les pistes de l'analyse de chaîne. Vue de l'extérieur, une telle transaction peut être interprétée comme un petit coinjoin entre deux personnes. Mais en réalité, tout comme la transaction Stonewall x2, il s'agit d'un paiement. Cette méthode génère donc des incertitudes dans l'analyse de chaîne, voire oriente vers de fausses pistes. Même si l'observateur extérieur parvient à identifier le pattern de la transaction Stonewall, il ne disposera pas de toutes les informations. Il ne pourra pas déterminer lequel des deux UTXOs de mêmes montants correspond au paiement. De plus, il ne pourra pas déterminer si les deux UTXOs en entrée proviennent de deux personnes différentes ou s'ils appartiennent à une seule personne qui les a fusionnés. Ce dernier point est dû au fait que les transactions Stonewall x2 suivent exactement le même pattern que les transactions Stonewall. Vu de l'extérieur et sans informations supplémentaires sur le contexte, il est impossible de différencier une transaction Stonewall d'une transaction Stonewall x2. Or, les premières ne sont pas des transactions collaboratives, alors que les secondes le sont. Cela permet d'ajouter encore plus de doutes sur cette dépense.

Termes associés : COINJOIN | STONEWALL X2

STONEWALL X2

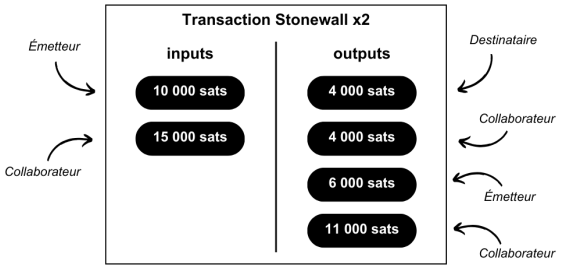
CONFIDENTIALITÉ

Forme spécifique de transaction Bitcoin visant à accroître la confidentialité des utilisateurs lors d'une dépense, par la collaboration avec une tierce personne non impliquée dans cette dépense. Cette méthode simule un mini-coinjoin entre deux participants, tout en effectuant un paiement à une troisième partie. Les transactions Stonewall x2 sont disponibles à la fois sur l'application Samourai Wallet et sur le logiciel Sparrow Wallet (les deux sont interopérables).

Son fonctionnement est relativement simple : on utilise un UTXO en notre possession pour effectuer le paiement et on sollicite l'aide d'une tierce personne qui contribue également avec un UTXO lui appartenant. La transaction se solde avec quatre outputs : deux d'entre eux de montants égaux, l'un destiné à l'adresse du bénéficiaire du paiement, l'autre à une adresse appartenant au collaborateur. Un troisième UTXO est renvoyé à une autre adresse du collaborateur, lui permettant de récupérer le montant initial (une action neutre pour lui,

modulo les frais de minage), et un dernier UTXO revient à une adresse nous appartenant, qui constitue le change du paiement. On définit ainsi trois rôles différents dans les transactions Stonewall x2 :

- L'émetteur, qui réalise le paiement effectif ;
- Le collaborateur, qui met des bitcoins à disposition afin d'améliorer l'ensemble d'anonymat de la transaction, tout en récupérant intégralement ses fonds à la fin ;
- Le destinataire, qui peut ignorer la nature spécifique de la transaction et attend simplement un paiement de la part de l'émetteur.



La structure Stonewall x2 ajoute énormément d'entropie à la transaction et vient brouiller les pistes de l'analyse de chaîne. Vue de l'extérieur, une telle transaction peut être interprétée comme un petit coinjoin entre deux personnes. Mais en réalité, il s'agit d'un paiement. Cette méthode génère donc des incertitudes dans l'analyse de chaîne, voire oriente vers de fausses pistes. Même si l'observateur extérieur parvient à identifier le pattern de la transaction Stonewall x2, il ne disposera pas de toutes les informations. Il ne pourra pas déterminer lequel des deux UTXOs de mêmes montants correspond au paiement. De plus, il ne sera pas en mesure de savoir qui a effectué le paiement. Enfin, il ne pourra pas déterminer si les deux UTXOs en entrée proviennent de deux personnes différentes ou s'ils appartiennent à une seule personne qui les a fusionnés. Ce dernier point est dû au fait que les transactions Stonewall classiques suivent exactement le même pattern que les transactions Stonewall x2. Vu de l'extérieur et sans informations supplémentaires sur le contexte, il est impossible de différencier une transaction Stonewall d'une transaction Stonewall x2. Or, les premières ne sont pas des transactions collaboratives, alors que les secondes le sont. Cela permet d'ajouter encore plus de doutes sur la dépense.

Termes associés : **COINJOIN** **STONEWALL**

STRATUM

↗ MINAGE

Protocole réseau superposé à Bitcoin spécifiquement conçu pour optimiser la communication entre les mineurs individuels et les serveurs des pools de minage afin d'augmenter leur rentabilité. Stratum a été annoncé fin 2012 par Marek Palatinus, plus connu sous le pseudonyme de « Slush » et fondateur de la toute première pool de minage, Slush Pool, aujourd'hui rebaptisée Braiins Pool.

Stratum est venu remplacer l'ancien protocole Getwork, alors devenu obsolète.

Il est important de comprendre que Stratum n'est pas intégré en tant que règle dans Bitcoin, mais est plutôt un logiciel spécifique utilisé par les pools. Bien que les pools de minage aient la liberté de ne pas l'utiliser, Stratum s'est imposé comme la référence pour le minage sur Bitcoin depuis plus de dix ans. Sa seconde version, Stratum V2, vise à perfectionner Stratum et à réduire les inquiétudes de centralisation associées aux pools de minage que ce dernier a engendrées.

Termes associés : **MINING POOL** | **STRATUM V2**

STRATUM V2

↗ MINAGE

Évolution de Stratum, le célèbre protocole réseau superposé à Bitcoin conçu pour le minage. Ce type de protocole est conçu pour optimiser la communication entre les mineurs individuels et les serveurs des pools de minage afin d'augmenter leur rentabilité. Développé par la pool Braiins (anciennement « Slush Pool »), Stratum V2 introduit plusieurs améliorations, notamment une communication plus efficace entre les mineurs et les pools de minage, réduisant ainsi la bande passante nécessaire. Il réduit également les besoins d'infrastructures pour les pools. En termes de sécurité, il ajoute une authentification cryptographique afin d'empêcher différentes attaques de l'homme du milieu, notamment les écoutes clandestines et la redirection malveillante du hashrate. Un aspect important de Stratum V2 est qu'il inclut des sous-protocoles permettant aux mineurs individuels de choisir leurs propres ensembles de transactions à inclure dans les blocs sur lesquels ils travaillent. Cette fonctionnalité donne plus de pouvoir aux mineurs individuels, contrairement au protocole original où les pools avaient un contrôle total sur le *block template*. Cette méthode permettrait ainsi de renforcer la décentralisation du processus de minage par les pools.

Termes associés : **MINING POOL** | **STRATUM**

STUCKLESS PAYMENT

↯ LIGHTNING NETWORK

Proposition visant à résoudre le problème des paiements bloqués sur le Lightning Network. Avec les HTLC actuels, un paiement peut rester coincé sans être ni complété ni annulé, par exemple si un nœud intermédiaire tombe hors ligne ou retient volontairement le HTLC. L'expéditeur doit alors attendre l'expiration du *timelock* pour récupérer ses fonds, et ne peut pas relancer le même paiement sans risquer un double paiement au destinataire.

Les *stuckless payments*, proposés par Hiroki Gondo, reposent sur les PTLC et ajoutent un tour de communication supplémentaire entre l'expéditeur et le destinataire lors de la mise en place et du règlement du paiement. Ce mécanisme introduit une clé de confirmation (*acknowledgment key*) que l'expéditeur contrôle : le destinataire ne peut finaliser le paiement qu'après avoir reçu cette clé. L'expéditeur peut ainsi lancer plusieurs tentatives en parallèle sur des routes différentes, avec la garantie que si l'une aboutit, toutes les autres échoueront automatiquement. Cela élimine le risque de double paiement et améliore

la fiabilité du routage sur Lightning.

Termes associés : HTLC | PTLC

SUBMARINE SWAP

LIGHTNING NETWORK | FR : ÉCHANGE SOUS-MARIN

Échange atomique entre des bitcoins on-chain et des bitcoins off-chain sur le Lightning Network, sans risque de contrepartie. Aucune des deux parties ne prend *custody* des fonds de l'autre : soit l'échange réussit intégralement, soit il échoue et chacun récupère ses fonds.

Le mécanisme repose sur les mêmes HTLC (*Hash Time-Locked Contracts*) que ceux utilisés dans le routage Lightning. Le destinataire du paiement Lightning génère une préimage secrète dont le hash sert à construire à la fois un contrat on-chain et un paiement off-chain. Le payeur on-chain verrouille ses bitcoins dans un script conditionnel : les fonds peuvent être réclamés par le destinataire s'il fournit la préimage et sa signature, ou récupérés par le payeur après expiration du *timelock*. Lorsque le paiement Lightning est effectué, la préimage est révélée, ce qui permet au destinataire de réclamer les fonds on-chain.

Termes associés : HTLC | LOOP

SUBVENTION DE BLOC

PROTOCOLE | EN : BLOCK SUBSIDY

Quantité de nouvelles unités pouvant être créées par le mineur qui résout un bloc. Cette subvention fait partie de la récompense de bloc avec les frais de transaction. Elle est distribuée au sein d'une transaction spécifique que l'on appelle « coinbase ». Initialement fixée à 50 bitcoins par bloc en 2009, cette subvention est réduite de moitié tous les 210 000 blocs (soit environ tous les quatre ans) grâce à un processus connu sous le nom de « halving ». Lorsque la subvention passera en dessous du montant de 1 sat, elle ne pourra plus être collectée, et la récompense de bloc reposera uniquement sur les frais de transaction. Sauf s'il y a une modification du protocole, la masse monétaire en circulation ne pourra alors plus être augmentée.

Termes associés : HALVING | RÉCOMPENSE DE BLOC

SURCOUCHE

COUCHE SUPÉRIEURE | EN : LAYER

Protocole ou réseau construit en supplément, en s'empilant sur le réseau Bitcoin principal. Elle utilise le réseau Bitcoin comme une fondation et est donc dépendante de son protocole. Cependant, le réseau Bitcoin n'est pas dépendant de ses surcouches. Un exemple d'une telle surcouche est le Lightning Network.

Ces surcouches sont conçues pour étendre les capacités du réseau Bitcoin en ajoutant des fonctionnalités ou des capacités supplémentaires, telles que des transactions plus rapides, des jetons ou des micropaiements. Elles sont souvent créées pour résoudre certaines limitations du réseau Bitcoin, tout en bénéficiant de sa sécurité et de sa décentralisation. Il est important de noter

que bien que ces surcouches soient construites sur le réseau Bitcoin, elles ont leurs propres protocoles et mécanismes distincts de ceux du réseau Bitcoin lui-même.

Termes associés : **LIGHTNING NETWORK** | **SIDECHAIN**

SURFACE D'ATTAQUE

‹/› INFORMATIQUE | EN : ATTACK SURFACE

Désigne l'ensemble des points d'entrée potentiels qu'un attaquant peut exploiter pour accéder à un système. La surface d'attaque inclut toutes les interfaces, les services, les ports, les protocoles, et les autres vecteurs de communication qui pourraient être exposés. Une surface d'attaque étendue augmente les vulnérabilités et rend donc le système plus susceptible d'être attaqué. Réduire la surface d'attaque est une bonne pratique de sécurité qui implique la désactivation des services non nécessaires. Par exemple, dans le cadre de Bitcoin, les *hardware wallets* ont une surface d'attaque beaucoup plus réduite que les *software wallets*. En effet, ils isolent les clés sur un dispositif non connecté à Internet, disposant de très peu d'interfaces et de services. À l'opposé, les *software wallets* sont installés sur des PC polyvalents, qui ont une surface d'attaque bien plus vaste en raison des nombreuses fonctionnalités et des connexions réseau qu'ils supportent.

Terme associé : **HARDWARE WALLET**

SWAP-IN POTENTIAM

LIGHTNING NETWORK

Protocole qui permet de transférer des bitcoins on-chain vers le Lightning Network de manière quasi instantanée, sans recourir aux canaux en 0-conf. Proposé par Jesse Posner et ZmnSCPxj en 2023, il repose sur une adresse on-chain partagée entre l'utilisateur (Alice) et un LSP (Bob).

Cette adresse encode deux chemins de dépense : un chemin coopératif, où Alice et Bob signent conjointement pour déplacer les fonds, et un chemin de récupération unilatérale, où Alice peut reprendre ses bitcoins après l'expiration d'un *timelock*. Lorsque des fonds sont reçus et confirmés sur cette adresse, Alice peut, avec l'aide du LSP, les basculer immédiatement dans un canal Lightning existant ou nouveau, via le chemin coopératif. Si le LSP est indisponible, Alice conserve la possibilité de récupérer ses fonds on-chain après le délai prévu.

L'avantage principal du *Swap-in Potentiam* est de réduire les exigences de confiance par rapport à d'autres méthodes d'ouverture instantanée de canaux. La copropriété temporaire des fonds et le *timelock* de sécurité garantissent que ni Alice ni le LSP ne peuvent voler les fonds. Le portefeuille Phoenix de la société ACINQ utilise une variante de ce protocole appelée « Swaproot », adaptée aux contraintes des terminaux mobiles dont la connectivité est intermittente.

Termes associés : **GOSSIP** | **LIGHTNING NETWORK**

SWAPROOT

LIGHTNING NETWORK

Protocole de dépôt on-chain vers un canal Lightning conçu par ACINQ et déployé dans le portefeuille Phoenix en février 2024. Swaproot est une évolution du protocole *swap-in potentiam* qui exploite Taproot et MuSig2 pour rendre les transactions de swap-in moins coûteuses, plus confidentielles et compatibles avec la rotation d'adresses.

L'ancien protocole utilisait une construction *pay-to-script* de la forme « clé utilisateur + clé serveur OU clé utilisateur + délai ». Cette structure était identifiable on-chain et imposait une adresse statique. Swaproot remplace le chemin coopératif par une dépense *key-path* Taproot : les clés de l'utilisateur et du serveur sont agrégées via MuSig2 en une seule clé publique. Lorsque les deux parties coopèrent pour effectuer un *splice-in*, la transaction est indiscernable d'un paiement P2TR ordinaire. Le chemin de remboursement (clé utilisateur + délai) est relégué dans le *script-path*, invisible sauf en cas de récupération unilatérale.

Termes associés : **MUSIG2** | **SWAP-IN POTENTIAM**

SWEEP TRANSACTION

CONFIDENTIALITÉ | FR : TRANSACTION DE BALAYAGE

Pattern ou modèle de transaction utilisé en analyse de chaîne pour déterminer la nature de la transaction. Ce modèle se caractérise par la consommation d'un seul UTXO en entrée et la production d'un seul UTXO en sortie. L'interprétation de ce modèle est que nous sommes en présence d'un auto-transfert. L'utilisateur s'est transféré ses bitcoins à lui-même, sur une autre adresse lui appartenant. Puisqu'aucun change n'existe sur la transaction, il est très peu probable que l'on soit en présence d'un paiement. En effet, lorsqu'un paiement est effectué, il est presque impossible que le payeur dispose d'un UTXO correspondant exactement au montant requis par le vendeur, en plus des frais de transaction. En général, le payeur est donc obligé de produire un output de change. Nous savons alors que l'utilisateur observé est vraisemblablement encore en possession de cet UTXO. Dans le cadre d'une analyse de chaîne, si nous savons que l'UTXO utilisé en input de la transaction appartient à Alice, on peut supposer que l'UTXO en output lui appartient également.

Termes associés : **ANALYSE DE CHAINE** | **UTXO**

SWISS BITCOIN PAY

OUTIL

Solution de terminal de paiement Bitcoin destinée aux commerçants et aux entreprises. Swiss Bitcoin Pay permet d'accepter des paiements en bitcoins, aussi bien *on-chain* que via le Lightning Network, à travers une application mobile ou une API intégrable dans les systèmes de caisse existants.

Le service fonctionne sur un modèle hybride : il facilite la réception des paiements sans que le commerçant ait besoin de gérer lui-même un nœud ou une infrastructure technique, puis il reverse les encaissements automatiquement chaque jour. Les fonds reçus peuvent être conservés en bitcoins, convertis automatiquement en monnaie fiat, ou bien un *mix* des deux, selon les préférences du commerçant.

Swiss Bitcoin Pay se positionne entre les solutions entièrement auto-hébergées comme BTCPay Server et les processeurs de paiement *custodians*, et offre un compromis entre simplicité d'utilisation et souveraineté sur les fonds.

Termes associés : **BTCPAY SERVER** | **LIGHTNING NETWORK**

SYBIL

ATTAQUE

Attaque informatique dans laquelle un individu ou une entité crée de multiples fausses identités dans un système pour exercer une influence indue ou gagner un avantage non autorisé. Dans le cadre de Bitcoin, une telle attaque vise à obtenir une influence indue sur le système, permettant ainsi de manipuler les mécanismes de consensus. Pour contrer les attaques Sybil, Satoshi Nakamoto a imaginé un système de preuve de travail qui impose un coût marginal non négligeable à la création de multiples votes.

Termes associés : **CONSENSUS** | **PREUVE DE TRAVAIL**

SYNONYM

ORGANISATION

Entreprise technologique fondée par John Carvalho, axée sur le développement de logiciels et de protocoles qui favorisent une économie pair-à-pair centrée sur Bitcoin. Synonym est à l'origine du protocole Slashtags, un standard web décentralisé qui permet de dériver des paires de clés supplémentaires à partir d'une graine BIP-0039 pour créer des profils, des contacts et une authentification sans mot de passe. L'entreprise a également développé Bitkit, un portefeuille mobile avec un nœud Lightning embarqué.

Terme associé : **LIGHTNING NETWORK**

T

TAIL EMISSION

PROTOCOLE

Mécanisme de politique monétaire consistant à maintenir indéfiniment une récompense de bloc fixe après l'épuisement du calendrier d'émission initial d'une cryptomonnaie. Contrairement au modèle de Bitcoin, où la subvention de bloc diminue progressivement par *halvings* jusqu'à atteindre zéro, une *tail emission* garantit un flux constant de nouvelles unités créées à chaque bloc. Bien que très controversée, il s'agit d'une proposition qui ressurgit parfois dans les débats autour de Bitcoin.

Monero est l'exemple le plus notable de cette approche : depuis juin 2022, chaque bloc produit une récompense fixe de 0,6 XMR, après l'épuisement de l'émission principale d'environ 18,1 millions d'unités.

Les partisans de la *tail emission* avancent qu'elle résout le problème du financement à long terme de la sécurité du réseau en garantissant une incitation permanente pour les mineurs, sans dépendre exclusivement des frais de transaction. Ils soutiennent aussi que la perte inévitable de pièces (clés oubliées, décès des détenteurs) compenserait naturellement la création monétaire, faisant converger l'offre en circulation vers un équilibre stable. Les opposants considèrent qu'elle contredit le principe d'une offre maximale fixe et déterministe, caractéristique fondamentale de Bitcoin. Ils arguent également que l'inflation permanente dilue la valeur des unités existantes et qu'un marché de frais de transaction suffisamment développé peut assurer seul la sécurité du réseau à long terme.

Dans le contexte de Bitcoin, l'ajout d'une *tail emission* nécessiterait un *hard fork* et constitue une proposition très controversée. La politique monétaire fixe de Bitcoin, avec un plafond de 21 millions de bitcoins, est considérée par une large part de la communauté comme un attribut fondamental du protocole.

Termes associés : **HALVING** | **RÉCOMPENSE DE BLOC**

TAPD

OUTIL

Logiciel daemon qui constitue l'implémentation de référence du protocole Taproot Assets, développé par Lightning Labs. `tapd` permet de créer (*mint*), envoyer et recevoir des actifs numériques encodés dans des sorties Taproot sur la blockchain Bitcoin.

Termes associés : **LND** | **TAPROOT ASSETS PROTOCOL**

TAPROOT

PROTOCOLE

Mise à jour majeure du protocole Bitcoin, adoptée par le biais d'un soft fork en novembre 2021. Cette mise à jour apporte des améliorations significatives en termes de confidentialité, d'efficacité et de flexibilité, en implémentant les BIP-0340, BIP-0341 et BIP-0342. Cette mise à jour a été verrouillée au bloc 687 284, le 12 juin 2021, lorsque 90 % des blocs générés pendant une période ont émis un signal favorable, manifestant ainsi la préparation des mineurs à activer la mise à jour (*Speedy Trial*). L'activation a finalement eu lieu au bloc 709 632, le 14 novembre 2021, soit presque quatre ans après les premières discussions à ce sujet entre Pieter Wuille, Andrew Poelstra et Gregory Maxwell. Ce fut la première tentative de mise à jour majeure depuis l'épineuse activation de SegWit en 2017.

Taproot est également le nom du BIP-0341, implémenté au sein du soft fork de même nom, qui introduit un nouveau modèle de script nommé P2TR. Un script P2TR verrouille des bitcoins sur une clé publique Schnorr unique, dénommée K. Cependant, cette clé K est en réalité un agrégat d'une clé publique P et d'une clé publique M, cette dernière étant calculée à partir de la racine de Merkle d'une liste de `scriptPubKey`. Les bitcoins verrouillés avec un script P2TR peuvent être dépensés de deux manières distinctes : soit en publiant une signature pour la clé publique P, soit en satisfaisant l'un des scripts contenus dans l'arbre de Merkle. La première option est appelée « *key path* » (chemin de clé) et la seconde « *script path* » (chemin de script).

Terme associé : **P2TR**

TAPROOT ASSETS PROTOCOL

COUCHE SUPÉRIEURE

Protocole développé par Lightning Labs permettant d'émettre des actifs sur la blockchain principale de Bitcoin, en tirant parti de la mise à jour Taproot. Taproot Assets permet la création d'actifs fongibles comme des stablecoins et non fongibles comme des NFT. Les Taproot Assets peuvent être transférés via des transactions Bitcoin classiques ou via le Lightning Network. Ce protocole utilise des *Merkle-Sum Sparse Merkle Trees* (MS-SMT), une sorte de combinaison des MST et des SMT, pour assurer la validité et l'audibilité des actifs.

Taproot Assets Protocol s'appelait « TARO » auparavant.

Termes associés : **LIGHTNING NETWORK** **TAPROOT**

TAPROOT CHANNEL

LIGHTNING NETWORK

Canal Lightning ancré dans une sortie Taproot (P2TR), plutôt que dans une sortie SegWit classique (P2WSH). Les *Taproot channels* tirent parti des améliorations apportées par le soft fork Taproot activé en novembre 2021 sur Bitcoin.

En cas de fermeture coopérative, un *Taproot channel* est plus efficace et plus confidentiel qu'un canal SegWit traditionnel : la transaction de fermeture ressemble à une dépense Taproot ordinaire via le *key path spend*, indiscernable d'un paiement simple on-chain. Cette propriété renforce la confidentialité des utilisateurs du Lightning Network.

Termes associés : **PTLC** **TAPROOT**

TAPSCRIPT

SCRIPT

Mise à jour qui a pour objet de modifier certains opcodes du langage de script classique de Bitcoin, afin de définir le nouveau langage de script utilisé pour les dépenses par chemin de script (*script path*) P2TR. Tapscript a été introduit par le BIP-0342, implémenté avec le soft fork Taproot.

Afin de mettre en œuvre les diverses modifications associées à Taproot, il s'est avéré nécessaire de revisiter le langage de script. C'est là l'objet de Tapscript qui désactive ou modifie certains opcodes, et vient en ajouter de nouveaux.

Terme associé : **TAPROOT**

TAPSIGNER

PORTEFEUILLE

Carte NFC de signature de transactions Bitcoin fabriquée par Coinkite. Le TapSigner stocke une clé privée maîtresse BIP-0032 et permet de signer des transactions en approchant la carte d'un téléphone compatible NFC, via un portefeuille comme Nunchuk. Dépourvue d'écran, cette carte constitue un dispositif de signature abordable, adapté aux petits montants ou à une utilisation en multisig.

Termes associés : **COINKITE** **NUNCHUK**

TARO

COUCHE SUPÉRIEURE

Ancien nom du protocole Taproot Assets Protocol.

Terme associé : **TAPROOT ASSETS PROTOCOL**

TCP

RÉSEAU

Sigle de « *Transmission Control Protocol* ». C'est un protocole de communication conçu pour assurer une transmission de données fiable sur Internet. Il établit une connexion, garantit l'ordre des données envoyées, gère la retransmission en cas de perte de paquets, et contrôle la congestion.

Termes associés : **I2P** **TOR**

TÉMOIN DE TRANSACTION

PROTOCOLE | EN : TRANSACTION WITNESS

Fait référence à une composante des transactions Bitcoin qui a été déplacée avec le soft fork SegWit afin de résoudre le problème de la malléabilité des transactions. Le témoin contient les signatures et les clés publiques nécessaires pour déverrouiller les bitcoins dépensés dans une transaction. Dans les transactions Legacy, le témoin représentait la somme des `scriptSig` de tous les inputs. Dans les transactions SegWit, le témoin représente la somme des `scriptWitness` de chaque input, et cette partie de la transaction est dorénavant déplacée dans un arbre de Merkle séparé au sein du bloc.

Avant SegWit, les signatures pouvaient être légèrement modifiées sans être invalidées avant qu'une transaction ne soit confirmée, ce qui changeait l'identifiant de la transaction. Cela rendait difficile la construction de divers protocoles, car une transaction non confirmée pouvait voir son identifiant changer. En séparant les témoins, SegWit rend les transactions non malléables, car tout changement dans les signatures n'affecte plus l'identifiant de la transaction (TXID), mais uniquement l'identifiant du témoin (WTXID). En plus de résoudre le problème de la malléabilité, cette séparation permet d'augmenter la capacité de chaque bloc.

Terme associé : **SEGWIT**

TEN31

ORGANISATION

Fonds de capital-risque spécialisé dans l'investissement dans les entreprises de l'écosystème Bitcoin. Fondé en 2020, Ten31 tire son nom du 31 octobre (10/31 en format américain), date de publication du White Paper par Satoshi Nakamoto en 2008.

TERMINAL CONSIGNEMENT

RGB

Dans le cadre du protocole RGB, le *Terminal Consignment* désigne le *transfer consignment* intégrant l'état final d'un contrat, obtenu par la *State Transition* générée à partir de l'*Invoice* du destinataire (*payee*). Ce mécanisme représente le point d'aboutissement d'un transfert, en rassemblant l'ensemble des données nécessaires pour attester que la propriété ou l'état du contrat a été dûment transmis et validé.

Termes associés : **CONSIGNMENT** **STATE TRANSITION**

TESTNET

PROTOCOLE

Version alternative de Bitcoin utilisée exclusivement à des fins de test et de développement. Il s'agit d'un réseau séparé du réseau principal (*mainnet*), avec ses propres blocs et transactions, permettant aux développeurs de tester de nouvelles fonctionnalités, applications et mises à jour sans risque pour le réseau principal. Le testnet permet également d'éviter de payer des frais de transaction lors de tests. Les bitcoins utilisés sur le testnet n'ont aucune valeur réelle.

Termes associés : **MAINNET** | **SIGNET**

TESTNET RESET

RÉSEAU

Opération consistant à relancer le réseau de test de Bitcoin avec un nouveau bloc genesis, effaçant ainsi l'historique et les soldes existants. Ce mécanisme garantit que les coins de test restent faciles à miner et sans valeur marchande, ce qui est important pour que le testnet remplisse sa fonction : offrir un environnement d'expérimentation gratuit où les développeurs peuvent tester des attaques, des bugs ou de nouvelles fonctionnalités sans conséquence financière.

Le testnet a été réinitialisé trois fois dans l'histoire de Bitcoin. Le premier testnet a été introduit en octobre 2010. Un premier reset a été réalisé en février 2011, afin de créer ce que l'on appellera rétrospectivement « testnet2 ». Gavin Andresen a ensuite lancé le « testnet3 » en avril 2012. Après 13 ans de service, le testnet3 a finalement été remplacé par le testnet4 en 2024, défini dans le BIP-0094, notamment parce que ses coins avaient commencé à acquérir une valeur marchande.

Termes associés : **TESTNET** | **TESTNET4**

TESTNET4

RÉSEAU

Quatrième itération du réseau de test public de Bitcoin, lancée en 2024 pour remplacer le testnet3 devenu dysfonctionnel après 13 ans de service. Testnet4 est un réseau distinct du *mainnet* où les bitcoins (TBTC) n'ont aucune valeur, permettant aux développeurs de tester leurs applications sans risque financier.

Testnet3 cumulait plusieurs problèmes. Un bug dans la règle de réinitialisation de la difficulté (qui la ramenait à 1 si aucun bloc n'avait été trouvé depuis 20 minutes) provoquait régulièrement des tempêtes de blocs, propulsant la hauteur de chaîne au-delà de 2,5 millions de blocs. La récompense de bloc était tombée à 0,014 TBTC, rendant le minage inefficace pour distribuer des coins de test. Et surtout, en 2024, des TBTC avaient commencé à être achetés et vendus pour des airdrops frauduleux, brisant le principe fondamental de leur absence de valeur.

Testnet4, défini dans le BIP-0094, corrige le mécanisme de difficulté en interdisant l'exception des 20 minutes pour le premier bloc de chaque période de retargeting et en basant le calcul de l'ajustement sur la difficulté du premier bloc

de la période précédente plutôt que sur celle du dernier. Il utilise le port 48333 et un nouveau bloc genesis. Il est pris en charge par Bitcoin Core à partir de la version 28.0.

Termes associés : **SIGNET** **TESTNET**

THE DAO

HISTOIRE

Projet lancé en 2016 sur Ethereum, qui visait à créer un fonds d'investissement autonome géré par des *smart contracts*. Bien que principalement lié à Ethereum, The DAO a eu un impact important sur l'évolution de Bitcoin. En juin 2016, The DAO a été victime d'un piratage massif. En réaction, la communauté Ethereum a réalisé un *hard fork*, qui s'est avéré être un échec retentissant. Cet incident a consolidé les positions conservatrices des *small blockers* durant la Blocksize War de Bitcoin. Ils ont utilisé cet événement pour argumenter contre les changements rapides dans le protocole Bitcoin, comme ceux prônés par les *big blockers*, et contre l'idée de réaliser des *hard forks*, comme cela a été le cas sur Ethereum suite au hack de The DAO.

Terme associé : **BLOCKSIZE WAR**

THE EYE OF SATOSHI

OUTIL

Implémentation open source de watchtower pour le Lightning Network. Une watchtower surveille en permanence la blockchain Bitcoin au nom d'un utilisateur et intervient automatiquement en diffusant une transaction de pénalité si une contrepartie tente de tricher en publiant un ancien état du canal.

Termes associés : **LIGHTNING NETWORK** **WATCHTOWER**

THUNDERHUB

OUTIL

Interface web open source de gestion et de monitoring de nœuds Lightning Network basés sur LND. L'outil permet de gérer les canaux de paiement, d'envoyer et de recevoir des paiements on-chain et Lightning, de rééquilibrer la liquidité et de consulter des rapports détaillés sur l'activité du nœud.

Termes associés : **CANAL DE PAIMENT** **NOEUD LIGHTNING**

TIDES

↗ MINAGE

Sigle de « *Transparent Index Of Distinct Extended Shares* ». C'est une méthode de calcul de la rémunération des mineurs dans le contexte des pools de minage introduite par la pool Ocean en 2023. Cette méthode répartit les récompenses en fonction d'un pourcentage pondéré du travail consacré aux shares les plus récemment trouvées. Chaque preuve est rémunérée plusieurs fois, avec un calcul de récompense incluant les frais de transaction. Ce système assure une grande précision dans les paiements des mineurs, sans nécessiter un intermédiaire de garde pour le traitement des paiements, contrairement à d'autres méthodes comme FPPS. TIDES est conçu pour des rémunérations transparentes.

Terme associé : **SHARES**

TIME-DILATION ATTACK

ATTAQUE

Attaque sur le Lightning Network qui combine une attaque éclipse avec un ralentissement délibéré de la livraison des blocs à la victime. L'attaquant prend le contrôle de toutes les connexions du nœud Bitcoin de la victime, puis lui transmet les nouveaux blocs avec un retard calculé (typiquement juste sous le seuil de détection de 30 minutes de Bitcoin Core). La victime accumule progressivement un retard par rapport à la chaîne réelle. Ce décalage temporel est exploité pour contourner les mécanismes de sécurité du Lightning Network qui reposent sur des *timelocks*. Par exemple, l'attaquant peut diffuser une ancienne transaction d'engagement révoquée sur le réseau réel. La victime, dont la vue de la blockchain est en retard, ne détecte pas cette tentative de fraude à temps pour publier une transaction de pénalité avant l'expiration du délai de contestation. L'attaque ne nécessite aucune puissance de minage.

Ce concept a été formalisé en 2020 par Antoine Riard et Gleb Naumenko. Trois variantes ont été identifiées, dont la plus efficace (attaque de finalisation de paquet) ne nécessite que quelques heures d'éclipse et un seul canal avec la victime. Les contre-mesures reposent sur l'utilisation de *watchtowers* sur une infrastructure indépendante, la réception de blocs par des canaux alternatifs et le renforcement des protections anti-éclipse.

Termes associés : **ECLIPSE** **HTLC** **WATCHTOWER**

TIME WARP

ATTAQUE | FR : DISTORSION TEMPORELLE

Attaque théorique sur le mécanisme d'ajustement de la difficulté de Bitcoin qui exploite une faille dans la manière dont les horodatages des blocs sont vérifiés à la frontière entre deux périodes de retarget. Tous les 2 016 blocs, le protocole recalcule la cible de difficulté en comparant l'horodatage du premier bloc de la période avec celui du dernier. L'attaque consiste, pour un mineur qui contrôle une majorité de la puissance de calcul, à manipuler ces horodatages : il fixe les horodatages de presque tous les blocs de la période au minimum autorisé par la règle *Median Time Past* (MTP), puis attribue au dernier bloc un horodatage dans le futur, qui respecte la règle du NAT + 2 heures. Le protocole en déduit alors que la période a duré bien plus longtemps qu'en réalité, ce qui entraîne une réduction artificielle et injustifiée de la difficulté pour la période suivante.

En répétant ce procédé sur plusieurs périodes successives, un attaquant pourrait théoriquement abaisser la difficulté jusqu'à produire des blocs à un rythme extrême (jusqu'à un maximum de 6 blocs par seconde). Cette décélération puis accélération extrême compromettrait la sécurité des fonds protégés par des *timelocks*, pourrait saturer la bande passante des nœuds du réseau, faciliterait certains types d'attaques comme les attaques éclipse, et entraînerait de nombreuses réorganisations majeures rendant les confirmations de transactions peu fiables. Pour l'attaquant, le principal bénéfice qu'il peut en tirer est de récupérer les subventions de blocs beaucoup plus rapidement que normalement.

La proposition de soft fork « Consensus Cleanup » (BIP-0054) vise à corriger cette vulnérabilité en imposant que le premier bloc d'une nouvelle période de difficulté porte un horodatage qui ne précède pas de plus de 2 heures celui du dernier bloc de la période précédente. Cette contrainte empêche la manipulation extrême des écarts temporels entre deux périodes de retarget et empêche donc cette attaque.

Notons que cette attaque est rendue possible par un bug historique (*off-by-one*) introduit par Satoshi Nakamoto dans le calcul de l'ajustement de la difficulté. Ce calcul prend pour base une durée de 2 semaines (censée correspondre à 2016×10 minutes), et la compare à la différence entre le premier et le dernier bloc de la période, afin de déterminer un écart. Le problème, c'est qu'entre le premier et le dernier bloc d'une période, il n'existe en réalité que 2015 intervalles, et non 2016. Pour que le calcul soit correct, il aurait fallu comparer avec la différence entre le dernier bloc de la période précédente et le dernier bloc de la période actuelle qui vient de se terminer, ce qui aurait bien donné 2016 intervalles. Outre le fait que ce bug biaise très légèrement le calcul d'ajustement d'environ 0,05 %, il est surtout ce qui rend possible les attaques de type time warp. En effet, si la comparaison portait sur le dernier bloc de la période précédente plutôt que sur le premier bloc de la période qui vient de se terminer, il y aurait un chevauchement des périodes, ce qui empêcherait ce type de manipulation. Le problème, c'est que pour corriger réellement ce bug historique, il faudrait faire un hard fork.

Termes associés : AJUSTEMENT DE LA DIFFICULTE | DIFFICULTÉ | MINAGE

TIMELOCK

PROTOCOLE | FR : VERROUILLAGE TEMPOREL

Primitive de contrat intelligent qui permet de définir une condition temporelle à remplir pour qu'une transaction puisse être ajoutée à un bloc. Il existe deux types de timelocks sur Bitcoin :

- Le timelock absolu, qui spécifie un moment précis avant lequel la transaction ne peut être incluse dans un bloc ;
- Le timelock relatif, qui définit un délai à partir de l'acceptation d'une transaction antérieure.

Le timelock peut être défini soit sous la forme d'une date exprimée en temps Unix, soit sous la forme d'un numéro de bloc. Enfin, le timelock peut s'appliquer soit à un output de transaction grâce à l'utilisation d'un opcode spécifique dans le script de verrouillage (`OP_CHECKLOCKTIMEVERIFY` ou `OP_CHECKSEQUENCEVERIFY`), soit à une transaction entière grâce à l'utilisation du champ `nLockTime`, soit à un input spécifique grâce à l'utilisation du champ `nSequence`.

On parle également parfois d'un « locktime » pour évoquer un timelock.

Termes associés : **NLOCKTIME** | **NSEQUENCE**

TLV

INFORMATIQUE

Format d'encodage de données *type-length-value* structuré en trois champs : un type identifiant la nature de l'information, une longueur indiquant la taille des données, et une valeur contenant les données elles-mêmes. Ce format permet de créer des messages extensibles où de nouveaux types de champs peuvent être ajoutés sans rompre la compatibilité avec les implémentations existantes.

Sur le Lightning Network, le TLV est utilisé pour structurer les messages du protocole BOLT, permettant d'inclure des informations optionnelles comme les métadonnées de paiement ou les données de routage. Dans le protocole Taproot Assets, chaque feuille de l'arbre d'actifs contient un blob TLV qui encode la version, l'*asset ID*, le montant détenu, ainsi que les signatures et données relatives aux transferts précédents de l'actif.

Termes associés : **LIGHTNING NETWORK** | **TAPROOT ASSETS PROTOCOL**

TOR

RÉSEAU

Réseau de serveurs relais (*nodes*) qui permet d'anonymiser l'origine des connexions TCP sur internet. Il fonctionne en encapsulant les données dans plusieurs couches de chiffrement. Chaque nœud de relais enlève une couche pour révéler l'adresse du nœud suivant, jusqu'à atteindre la destination finale. Le réseau Tor assure l'anonymat en empêchant les nœuds intermédiaires de connaître à la fois l'origine et la destination des données, ce qui rend très difficile pour un observateur de retracer l'activité de l'utilisateur. Le réseau Tor peut être utilisé dans le cadre de Bitcoin pour éviter d'associer son adresse IP à un nœud Bitcoin, et donc éviter de faire fuiter certaines informations personnelles.

« Tor » signifie « The Onion Router ». Il est important de différencier Tor le réseau, et Tor Browser, un navigateur web établi sur Firefox qui est spécialement conçu pour utiliser le réseau Tor.

Termes associés : **I2P** | **NOEUD**

TPRV

PORTEFEUILLE

Préfixe de clé privée étendue pour les comptes Legacy et SegWit V1 sur Bitcoin Testnet.

Terme associé : **CLÉ ÉTENDUE**

TPUB

PORTEFEUILLE

Préfixe de clé publique étendue pour les comptes Legacy et SegWit V1 sur Bitcoin Testnet.

Terme associé : **CLÉ ÉTENDUE**

TRAMPOLINE ROUTING

LIGHTNING NETWORK | FR : ROUTAGE TRAMPOLINE

Méthode de routage Lightning qui permet à un nœud léger de déléguer la construction du chemin de paiement à un ou plusieurs nœuds intermédiaires spécialisés appelés *trampoline nodes*. C'est particulièrement utile pour les nœuds embarqués mobiles qui ne peuvent pas synchroniser le graphe complet du réseau en raison de contraintes de stockage, de puissance de calcul ou de connectivité intermittente.

L'expéditeur construit un « oignon trampoline » qui identifie les nœuds trampoline par leur identifiant complet. Il route le paiement vers le premier trampoline, qui calcule lui-même la route jusqu'au trampoline suivant ou jusqu'au destinataire. Les nœuds non trampoline sur la route ne nécessitent aucune mise à jour : ils relaient les HTLC comme des paiements classiques. Pour préserver la confidentialité, l'expéditeur peut chaîner au moins deux nœuds trampoline, de sorte qu'aucun d'entre eux ne sache s'il route vers le destinataire final ou vers un autre trampoline. Les nœuds trampoline facturent des frais supplémentaires pour ce service de *pathfinding*. Le routage trampoline peut être combiné avec les *blinded paths* pour masquer l'identité du destinataire, offrant ainsi confidentialité complète même depuis un appareil mobile.

Termes associés : **BLINDED PATHS** | **ROUTAGE EN OIGNON**

TRANSACTION COLLABORATIVE

CONFIDENTIALITÉ | EN : COLLABORATIVE TRANSACTION

Transaction Bitcoin qui implique plusieurs entités différentes en input de la transaction. Il s'agit donc également d'un modèle de transaction qui peut être utilisé en analyse de chaîne. L'exemple typique d'une transaction collaborative est le coinjoin, où plusieurs utilisateurs regroupent leurs UTXOs en inputs pour récupérer la somme (moins les frais de transaction) dans des outputs de même montant, afin d'empêcher le traçage en rompant l'historique des pièces.

Terme associé : **COINJOIN**

TRANSACTION D'ENGAGEMENT

LIGHTNING NETWORK | EN : COMMITMENT TRANSACTION

Dans le contexte d'un canal bidirectionnel au sein de Lightning, la transaction d'engagement est une transaction que les deux parties créent et signent, sans toutefois la publier sur la chaîne principale. Elle représente l'état actuel de la répartition des fonds entre les parties d'un canal, chaque paiement Lightning résultant en une nouvelle transaction d'engagement. Ces transactions sont valides, mais ne sont diffusées que lorsque le canal est clôturé unilatéralement. Elles contiennent des sorties pour chaque partie, reflétant la répartition des fonds selon les paiements Lightning effectués depuis l'ouverture du canal. Des mécanismes de pénalité sont associés pour dissuader les parties de diffuser des états obsolètes du canal, c'est-à-dire de vieilles transactions d'engagement qui reflètent une mauvaise répartition des fonds.

Termes associés : **CANAL DE PAIMENT** | **FORCE CLOSE**

TRANSACTION DE FINANCEMENT

LIGHTNING NETWORK | EN : FUNDING TRANSACTION

Transaction Bitcoin on-chain qui marque l'ouverture d'un canal de paiement sur le Lightning Network. Elle envoie des fonds vers une adresse multisignature 2/2 contrôlée conjointement par les deux parties du canal. Ces fonds constituent la capacité totale du canal et ne peuvent être dépensés que si les deux participants signent ensemble.

Avant de publier la transaction de financement sur la blockchain, l'initiateur du canal construit d'abord une transaction d'engagement (*commitment transaction*) signée par les deux parties. Cette précaution garantit que l'initiateur pourra récupérer ses fonds même si son pair devient non coopératif. Le canal est considéré comme ouvert une fois que la transaction de financement est confirmée. Son identifiant de sortie (`txid:output_index`) constitue le *channel point*, qui sert de référence unique pour identifier le canal durant toute sa durée de vie.

Termes associés : **CHANNEL POINT** | **TRANSACTION D'ENGAGEMENT**

TRANSACTION DE NOTIFICATION

CONFIDENTIALITÉ | EN : NOTIFICATION TRANSACTION

Transaction Bitcoin spécifique utilisée dans le cadre du protocole BIP-0047 (codes de paiement réutilisables). La transaction de notification constitue la première étape nécessaire pour établir un lien de paiement entre deux utilisateurs via leurs codes de paiement respectifs.

Concrètement, lorsque Alice souhaite envoyer des paiements à Bob en utilisant son code de paiement, elle doit d'abord lui envoyer une transaction de notification. Cette transaction contient, dans un output `OP_RETURN`, le code de paiement d'Alice chiffré à l'aide d'un secret partagé calculé avec ECDH (*Elliptic-Curve Diffie-Hellman*). Bob, en surveillant son adresse de notification, peut déchiffrer cette information et découvrir le code de paiement d'Alice. Il peut alors dériver les adresses correspondantes pour identifier les paiements futurs d'Alice.

La transaction de notification n'a besoin d'être envoyée qu'une seule fois pour chaque paire d'utilisateurs. Les paiements suivants entre Alice et Bob ne nécessitent plus cette étape.

Termes associés : **BIP-0047** | **PAYNYM**

TRANSACTION DE PÉNALITÉ

LIGHTNING NETWORK | EN : PENALTY TRANSACTION

Transaction Bitcoin utilisée sur le Lightning Network pour punir un pair qui a publié une ancienne transaction d'engagement révoquée sur la blockchain. Elle permet à la partie lésée de récupérer la totalité des fonds du canal, y compris la part de l'attaquant.

Le mécanisme repose sur les clés de révocation. Lorsqu'un état du canal est mis à jour, chaque partie remet à l'autre une clé qui permet de dépenser l'intégralité des fonds si l'ancien état venait à être publié. Si un nœud tente de tricher en diffusant un vieil engagement, son pair dispose d'un délai pour détecter la fraude et construire la transaction de pénalité à l'aide de cette clé de révocation. Ce délai est garanti par un timelock `OP_CHECKSEQUENCEVERIFY` sur la sortie du tricheur.

Les *watchtowers* sont des services tiers qui surveillent la blockchain en permanence pour détecter ces fraudes et diffuser les transactions de pénalité au nom des nœuds hors ligne.

Termes associés : **CHANNEL BREACH** | **FORCE CLOSE**

TRANSACTION NON CONFIRMÉE

PROTOCOLE | EN : UNCONFIRMED TRANSACTION

Transaction Bitcoin qui a été diffusée sur le réseau, mais n'a pas encore été incluse dans un bloc par un mineur. Elle reste dans les mempools des nœuds, en attente d'avoir des confirmations. La confirmation se produit lorsque la transaction est intégrée dans un bloc valide ajouté à la blockchain. Le temps nécessaire pour qu'une transaction soit confirmée dépend des frais de transaction payés et de la congestion du système.

Termes associés : **CONFIRMATION** | **MEMPOOL**

TRANSACTION PINNING

ATTAQUE | FR : ÉPINGLAGE DE TRANSACTION

Attaque qui consiste à rendre prohibitivement coûteux le remplacement d'une transaction en exploitant les protections des nœuds contre le gaspillage de bande passante, de CPU et de mémoire. L'attaquant détourne les règles prévues pour empêcher les abus du système de relais et les utilise pour bloquer les tentatives légitimes d'augmentation des frais dans un contexte multipartite.

Deux vecteurs principaux sont exploités. Le premier abuse de la règle 3 du BIP-0125, qui exige que la transaction de remplacement paie des frais absolus supérieurs à ceux de la transaction originale et de tous ses enfants. L'attaquant attache une transaction enfant volumineuse à faible taux de frais, ce qui force la victime à payer des frais démesurément élevés pour effectuer un remplacement par RBF. Le second vecteur exploite les limites de descendants : les nœuds restreignent les transactions à 101 000 vbytes de descendants ou 25 ancêtres/descendants au total. Un attaquant peut saturer ces limites avec ses propres transactions enfants, ce qui bloque complètement le CPFP.

Cette attaque représente une menace sérieuse pour le Lightning Network et les autres protocoles multipartites, où la gestion des frais est cruciale pour respecter les fenêtres temporelles de règlement. Plusieurs contre-mesures ont été développées : le *CPFP carve-out*, les *ephemeral anchors*, les transactions v3 et le *package relay*.

Termes associés : **EPHEMERAL ANCHORS** | **GOSSIP** | **PACKAGE RELAY**

TRANSACTION STANDARD

PROTOCOLE | EN : STANDARD TRANSACTION

Transaction Bitcoin qui, en plus de respecter les règles de consensus, entre également dans les règles de standardisation configurées par défaut sur les nœuds Bitcoin Core. Ces règles de standardisation sont imposées individuellement par chaque nœud Bitcoin, en plus des règles de consensus, pour définir la structure des transactions non confirmées qu'il accepte dans sa mempool et diffuse à ses pairs.

Ces règles sont donc configurées et exécutées en local par chaque nœud et peuvent varier d'un nœud à l'autre. Elles s'appliquent exclusivement sur les transactions non confirmées. Ainsi, un nœud n'acceptera une transaction qu'il jugerait non standard que si celle-ci est déjà incluse dans un bloc valide.

Notons que la majorité des nœuds laissent les configurations par défaut telles que préétablies dans Bitcoin Core, engendrant de fait une homogénéité des règles de standardisation à travers le réseau. Une transaction qui, bien que conforme aux règles de consensus, ne respecte pas ces règles de standardisation, aura des difficultés à se propager sur le réseau. Elle pourra toutefois être incluse dans un bloc valide si jamais elle atteint un mineur. Dans la pratique, ces transactions, qualifiées de non standard, sont souvent transmises directement à un mineur par des voies externes au réseau pair-à-pair de Bitcoin. C'est souvent le seul moyen pour confirmer ce type de transaction. Par exemple, une transaction qui n'alloue aucuns frais est à la fois valide selon les règles de consensus et non standard, car la politique par défaut de Bitcoin Core pour le paramètre `minRelayTxFee` est de `0.000001` (en BTC/kvB).

Termes associés : **BITCOIN CORE** | **MEMPOOL**

TRANSITION BUNDLE

RGB

Dans le cadre du protocole RGB, un *Transition Bundle* désigne un regroupement de *State Transitions* appartenant à un même contrat, toutes engagées simultanément dans une unique *witness transaction* Bitcoin. Ce mécanisme permet de consolider plusieurs mises à jour ou transferts dans un seul point d'ancrage *on-chain*, afin d'optimiser l'utilisation de l'espace de bloc.

Termes associés : **STATE TRANSITION** | **WITNESS TRANSACTION**

TREZOR

PORTEFEUILLE

Marque de *hardware wallets* fabriqués par la société tchèque SatoshiLabs, lancée en 2014 avec le Trezor One, considéré comme le premier portefeuille matériel Bitcoin commercialisé. Trezor propose des dispositifs qui permettent de sécuriser des clés privées hors ligne. Les appareils fonctionnent avec un *firmware open source*, ce qui permet la vérification indépendante du code par la communauté. L'interaction avec le portefeuille se fait via l'application Trezor Suite ou des logiciels compatibles comme Sparrow Wallet.

Termes associés : **GRAINE** | **HARDWARE WALLET**

TRIMMED HTLC

LIGHTNING NETWORK

Sur le Lightning Network, un HTLC est dit « *trimmed* » lorsque sa valeur est trop faible pour justifier la création d'un output dédié dans la transaction d'engagement. Concrètement, si le montant du HTLC est inférieur au *dust limit* du canal, il serait antiéconomique de le résoudre *on-chain*. Au lieu de créer un output séparé, la valeur de ces HTLC est absorbée dans les frais de transaction de la transaction d'engagement.

Ce mécanisme est indispensable pour permettre aux canaux Lightning de relayer des paiements très petits, potentiellement d'un seul satoshi, sans être limités par le seuil de poussière imposé par le protocole Bitcoin. Du point de vue opérationnel, les *trimmed HTLC* sont construits et résolus exactement comme les HTLC classiques : ils transitent normalement entre les nœuds et sont honorés ou annulés selon les mêmes règles. La différence n'apparaît que si la transaction d'engagement doit être publiée *on-chain*, auquel cas leur valeur est perdue dans les frais au lieu d'être récupérable.

Cette conception pose toutefois des problèmes de sécurité. Un acteur malveillant peut créer de nombreux *trimmed HTLC* sans intention de les résoudre, ce qui force sa contrepartie à payer des frais de transaction plus élevés, ou détruit une partie de la valeur du canal. Plusieurs améliorations ont été proposées, comme la redirection de la valeur des *trimmed HTLC* vers des *anchor outputs* éphémères ou l'utilisation de paiements probabilistes.

Termes associés : **ANCHOR OUTPUTS** | **HTLC** | **LIGHTNING NETWORK**

TUMBLEBIT

CONFIDENTIALITÉ

Concept de hub de paiement anonyme compatible avec Bitcoin proposé en 2016 par Ethan Heilman, Leen AlShenibr, Foteini Baldimtsi, Alessandra Scafuro et Sharon Goldberg. TumbleBit est un système de mixage de bitcoins qui ne requiert pas la confiance en un intermédiaire. Il permet à des utilisateurs de réaliser des paiements rapides, anonymes et hors-chaîne via un coordinateur appelé le *Tumbler*. TumbleBit garantit l'anonymat en s'assurant que même le *Tumbler* ne peut pas lier le paiement d'un payeur à son bénéficiaire. Le protocole TumbleBit assure que le *Tumbler* ne peut ni voler des bitcoins, ni imprimer de faux bitcoins en s'émettant des paiements à lui-même. L'anonymat offert par TumbleBit est comparable à celui d'un système *eCash* de Chaum. Cependant, ce concept n'a jamais été largement adopté, les techniques de confidentialité telles que le coinjoin chaumien lui étant préférées.

Terme associé : **COINJOIN**

TURBO CHANNEL

LIGHTNING NETWORK

Canal Lightning utilisable immédiatement, sans attendre les confirmations on-chain habituelles. Normalement, un canal Lightning n'est considéré comme opérationnel qu'après plusieurs confirmations de sa transaction de financement, soit généralement plusieurs dizaines de minutes. Un *turbo channel*, aussi appelé canal *zero-conf*, supprime cette attente : les deux extrémités du canal le considèrent comme actif dès la diffusion de la transaction de financement dans les mempools.

Ce mécanisme implique un compromis de confiance : tant que la transaction de financement n'est pas confirmée, la partie qui a ouvert le canal pourrait théoriquement effectuer une double dépense en publiant une transaction concurrente. En pratique, cette confiance est acceptée dans un cadre où l'ouvreur du canal est un fournisseur de services de liquidité (LSP) dont la réputation serait détruite par une telle fraude. Certaines implémentations renforcent la sécurité en exigeant que la transaction d'ouverture provienne d'une adresse multisignature avec un cosignataire de confiance.

Les *turbo channels* sont largement utilisés par les nœuds embarqués sur mobile comme Phoenix, car ça permet aux nouveaux utilisateurs d'envoyer et recevoir des paiements dès l'ouverture du canal, avec une expérience plus proche de celle des solutions custodiales.

Termes associés : CANAL DE PAIMENT | DOUBLE DÉPENSE

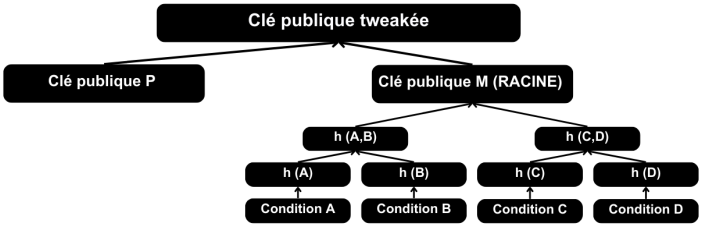
TWEAK

CRYPTOGRAPHIE

Dans le domaine de la cryptographie, « tweaker » une clé publique consiste à modifier cette clé en utilisant une valeur additive appelée le « tweak » de telle sorte qu'elle reste utilisable avec la connaissance de la clé privée d'origine et du tweak. Techniquement, un tweak est une valeur scalaire qui est ajoutée à la clé publique initiale. Si P est la clé publique et t est le tweak, la clé publique tweaked devient :

$$P' = P + tG$$

Où G est le générateur de la courbe elliptique utilisée. Cette opération permet d'obtenir une nouvelle clé publique dérivée de la clé originale tout en conservant certaines propriétés cryptographiques permettant de l'utiliser. Par exemple, on utilise cette méthode pour les adresses Taproot (P2TR) afin de pouvoir dépenser soit en présentant une signature Schnorr de façon traditionnelle, soit en remplissant l'une des conditions énoncées dans un arbre de Merkle, également appelé « MAST ».



Termes associés : P2TR | SCHNORR

TWENTYTWO DEVICES

ORGANISATION

Entreprise fondée par Alekos Filini qui développe le Portal, un *hardware wallet* au format plat qui communique avec les *smartphones* exclusivement via NFC, sans câble ni batterie. Le *firmware* du Portal est entièrement écrit en Rust et repose sur BDK. Le matériel comme le logiciel sont intégralement *open source*.

Terme associé : HARDWARE WALLET

TX - TRANSACTION

PROTOCOLE

Dans le contexte de Bitcoin, une transaction (abrégée « TX ») est une opération enregistrée sur la blockchain qui transfère la propriété de bitcoins d'une ou plusieurs entrées (inputs) vers une ou plusieurs sorties (outputs). Chaque transaction consomme des UTXOs en entrées, qui sont des outputs de transactions précédentes, et crée de nouveaux UTXOs en sorties, qui peuvent être utilisés comme entrées dans des transactions futures.

Chaque entrée comporte une référence à un output existant ainsi qu'un script de signature (`scriptSig`) qui remplit les conditions de dépense (`scriptPubKey`) établies par l'output auquel il fait référence. C'est ce qui permet de débloquer des bitcoins. Les outputs définissent de nouvelles conditions de dépense (`scriptPubKey`) pour les bitcoins transférés, souvent sous la forme d'une clé publique ou d'une adresse à laquelle les bitcoins sont maintenant associés.

Termes associés : SCRIPTPUBKEY | UTXO

TX0

CONFIDENTIALITÉ

Transaction préparatoire spécifique au protocole Whirlpool, qui constitue la première étape avant les cycles de coinjoin. La Tx0 prend un ou plusieurs UTXOs du portefeuille de l'utilisateur et les restructure en plusieurs outputs calibrés pour correspondre exactement aux montants de la pool de coinjoin choisie, en plus des frais de minage pour le premier cycle.

Concrètement, la Tx0 produit plusieurs types d'outputs : des UTXOs de montant fixe destinés à entrer dans les cycles de coinjoin (qui iront dans le compte premix), un output pour les frais du coordinateur, et éventuellement un output de change (le « *doxxic change* ») qui correspond à l'excédent qui ne peut pas être intégré dans la pool.

Termes associés : **POOL DE COINJOIN** | **WHIRLPOOL**

TXID - TRANSACTION IDENTIFIER

PROTOCOLE | FR : IDENTIFIANT DE TRANSACTION

Identifiant unique associé à chaque transaction Bitcoin. Il est généré en calculant le hachage `SHA256d` des données de la transaction. Le TXID sert de référence pour retrouver une transaction spécifique au sein de la blockchain. Il est également utilisé pour faire référence à un UTXO, via un *outpoint* qui est essentiellement la concaténation du TXID d'une transaction précédente et de l'index de l'output désigné (également appelé « *vout* »). Pour les transactions post-SegWit, le TXID ne prend plus en compte le témoin de la transaction, ce qui permet de supprimer la malléabilité.

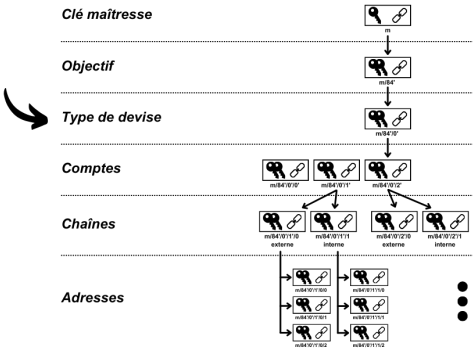
Terme associé : **WTXID**

TYPE DE DEVISE

PORTEFEUILLE | EN : COIN TYPE

Dans le cadre des portefeuilles déterministes et hiérarchiques (HD), le type de devise (*coin type* en anglais) est un niveau de dérivation qui permet de différencier les branches du portefeuille en fonction des différentes cryptomonnaies utilisées. Cet étage de dérivation, défini par le BIP-0044, se situe en profondeur 2 de la structure de dérivation, après la clé maîtresse et l'objectif (*purpose*). Par exemple, pour Bitcoin, l'index attribué est `0×80000000`, noté `/0'/` dans le chemin de dérivation. Cela signifie que toutes les adresses et comptes dérivés de ce chemin sont associés à Bitcoin. Cet étage de dérivation permet de bien séparer les différents actifs dans un portefeuille multi-devises. Voici les index utilisés pour différentes cryptomonnaies :

- Bitcoin : `0×80000000` ;
- Bitcoin Testnet : `0×80000001` ;
- Litecoin : `0×80000002` ;
- Dogecoin : `0×80000003` ;
- Ethereum : `0×8000003c`...



Terme associé : **CHEMIN DE DÉRIVATION**

U

UASF

PROTOCOLE

Sigle de « *User-Activated Soft Fork* ». Qualifie un soft fork dans Bitcoin lorsqu'il est initié et appliqué par les utilisateurs du réseau via leurs nœuds, sans dépendre de l'approbation des mineurs. Les nœuds du réseau mettent à jour leur logiciel pour adopter les nouvelles règles du soft fork et advienne que pourra !

Typiquement utilisé en cas d'urgence, notamment lorsque les mineurs sont majoritairement opposés à l'adoption d'un soft fork, l'UASF sert de moyen de pression pour éviter une concentration excessive de pouvoir chez les mineurs. Dans les faits, l'UASF est même devenu un outil de dissuasion, agité par les opérateurs de nœuds lorsque les mineurs abusent de leur pouvoir. Toutefois, si l'UASF est réellement appliqué, il présente des risques, notamment la possibilité d'une scission de la blockchain, créant une nouvelle chaîne qui peut manquer de valeur économique et de sécurité. La première proposition formelle d'UASF provient du développeur Shaolin Fry, qui a poussé le BIP-0148 en mars 2017 pour faire pression sur les mineurs qui refusaient de signaler SegWit.

Termes associés : **MASF** **SOFT FORK**

UDP

RÉSEAU

Protocole de communication utilisé sur Internet qui permet l'envoi de messages (datagrammes) entre ordinateurs sans établir de connexion préalable (contrairement à TCP). UDP est une méthode de transfert rapide, mais sans garantie de livraison, d'ordre des paquets, ou de gestion d'erreur. On l'utilise plutôt pour des applications nécessitant une diffusion rapide et en temps réel. Ce protocole avait été utilisé au sein du projet FIBRE (*Fast Internet Bitcoin Relay Engine*) pour accélérer la propagation de blocs Bitcoin.

Terme associé : **FIBRE**

UMBREL

OUTIL

Système d'exploitation pour serveurs personnels permettant d'auto-héberger des applications, dont un nœud Bitcoin et un nœud Lightning. Umbrel propose une interface web de type « *plug-and-play* » qui simplifie l'installation et la gestion d'un nœud complet, sans nécessiter de compétences techniques avancées.

Umbrel fonctionne sur du matériel dédié (Umbrel Home, Umbrel Pro) ou sur n'importe quel ordinateur compatible via umbrelOS. Son app store intègre des dizaines d'applications auto-hébergées : Bitcoin Core, LND, Core Lightning, Nextcloud, Tailscale, relais Nostr, et d'autres.

Termes associés : **LND** **START9**

UNIQUE ASSETS

COUCHE SUPÉRIEURE | FR : ACTIFS UNIQUES

Actifs émis via le protocole Taproot Assets qui sont indivisibles et non interchangeables. Contrairement aux actifs fongibles (comme une monnaie), un actif unique ne peut pas être divisé en fractions, et chaque unité est distincte des autres. Ils sont comparables aux NFT (*non-fungible tokens*) d'autres protocoles. Les *unique assets* peuvent représenter des certificats, des titres de propriété, des objets de collection numériques ou tout élément nécessitant une identification individuelle sur la blockchain Bitcoin.

Termes associés : **ASSET SPLIT** | **TAPROOT ASSETS PROTOCOL**

UNIVERSE

COUCHE SUPÉRIEURE

Dépôt de données servant de registre pour les actifs Taproot Assets. Un *universe* stocke et distribue les métadonnées, les preuves et l'historique de provenance des actifs émis sur Bitcoin via le protocole Taproot Assets. Les détenteurs et les acquéreurs potentiels d'actifs consultent un *universe* pour vérifier l'authenticité et la validité des actifs avant de les accepter. Chaque émetteur d'actifs peut maintenir son propre *universe*, et ces dépôts peuvent être synchronisés entre eux pour assurer une diffusion large des informations.

Termes associés : **TAPROOT ASSETS PROTOCOL** | **UNIQUE ASSETS**

UNIX

INFORMATIQUE

Famille de systèmes d'exploitation multitâche développée à partir de 1969 par Bell Labs. Conçu pour être simple, flexible et portable, UNIX a influencé de nombreux systèmes modernes. Ses principes fondamentaux incluent une structure de fichiers hiérarchique, l'utilisation de scripts shell et des utilitaires modulaires. UNIX est à l'origine de nombreuses variantes, dont Linux, et reste une référence en matière de stabilité et de performance.

Termes associés : **MTP - MEDIAN TIME PAST** | **NLOCKTIME**

UPRV

PORTEFEUILLE

Préfixe de clé privée étendue pour les comptes Nested SegWit sur Bitcoin Testnet.

Terme associé : **CLÉ ÉTENDUE**

UPUB

PORTEFEUILLE

Préfixe de clé publique étendue pour les comptes Nested SegWit sur Bitcoin Testnet.

Terme associé : **CLÉ ÉTENDUE**

URI

</> INFORMATIQUE

Sigle de « *Uniform Resource Identifier* ». C'est un format de chaîne de caractères standardisé utilisé pour identifier une ressource sur Internet. Un URI peut être soit un URL (*Uniform Resource Locator*), qui fournit un moyen d'accéder à une ressource en indiquant son emplacement sur un réseau informatique, soit un URN (*Uniform Resource Name*), qui nomme la ressource sans indiquer comment la localiser. Les URI sont importants dans le fonctionnement du *World Wide Web*, car ils permettent d'accéder à des ressources comme des pages web, des documents et des services.

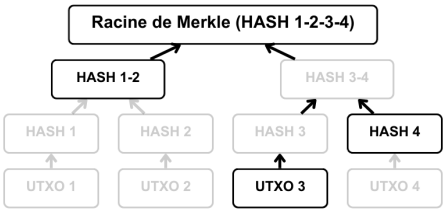
Dans le contexte de Bitcoin, un URI est utilisé spécifiquement pour faciliter les transactions. Il permet d'encoder une adresse de réception, ainsi que d'autres paramètres d'une transaction comme le montant, dans un format standardisé selon le BIP-0021. Cela simplifie le processus de paiement en permettant aux utilisateurs de cliquer sur un lien ou de scanner un code QR, qui intègre automatiquement les informations nécessaires dans leur application de portefeuille Bitcoin.

Termes associés : **ADRESSE DE RÉCEPTION** | **BIP-0021**

UTREEXO

PROTOCOLE

Protocole conçu par Tadge Dryja pour compacter l'UTXO set des nœuds Bitcoin à l'aide d'un accumulateur établi sur des arbres de Merkle. Contrairement à l'UTXO set classique qui nécessite un espace de stockage important, Utreexo réduit drastiquement la mémoire requise en ne stockant que les racines des arbres de Merkle. Cela permet au nœud de vérifier l'existence des UTXOs utilisés en entrées de transactions, sans avoir à conserver l'ensemble complet des UTXOs. En utilisant Utreexo, chaque nœud conserve uniquement des empreintes cryptographiques appelées racines de Merkle. Lorsqu'une transaction est effectuée, l'utilisateur fournit les preuves d'inclusion des UTXOs et les chemins de Merkle correspondants. Le nœud peut ainsi vérifier les transactions sans stocker tout l'UTXO set. Prenons un exemple avec un schéma pour comprendre ce mécanisme :



Dans cet exemple, j'ai intentionnellement réduit l'UTXO set à 4 UTXOs pour faciliter la compréhension. En réalité, il faut imaginer qu'il existe presque 140 millions d'UTXOs sur Bitcoin à l'heure où j'écris ces lignes. Sur ce schéma, le nœud Utreexo devrait uniquement conserver en RAM la Racine de Merkle. S'il

reçoit une transaction dépensant l'UTXO n° 3 (en noir), la preuve consisterait en les éléments suivants :

- L'UTXO 3 ;
- Le HASH 4 ;
- Le HASH 1-2.

Grâce à ces informations transmises par l'émetteur de la transaction, le nœud Utreexo effectue les vérifications suivantes :

- Il calcule l'empreinte de l'UTXO 3, ce qui lui donne HASH 3 ;
- Il concatène HASH 3 avec HASH 4 ;
- Il calcule leur empreinte, ce qui lui donne HASH 3-4 ;
- Il concatène HASH 3-4 avec HASH 1-2 ;
- Il calcule leur empreinte, ce qui lui donne la racine de Merkle.

Si la racine de Merkle qu'il obtient par son processus est la même que la racine de Merkle qu'il stockait dans sa RAM, alors il est persuadé que l'UTXO n° 3 fait bien partie de l'UTXO set.

Cette méthode permet de réduire les besoins en RAM pour les opérateurs de nœuds complets. Cependant, Utreexo présente des limites, notamment l'augmentation de la taille des blocs en raison des preuves supplémentaires et la dépendance potentielle des nœuds Utreexo envers les *Bridge Nodes* pour obtenir les preuves manquantes. Les *Bridge Nodes* sont des nœuds complets traditionnels qui fournissent les preuves nécessaires aux nœuds Utreexo, permettant ainsi une vérification complète. Cette approche offre un compromis entre efficacité et décentralisation, facilitant l'accès à la validation des transactions pour des utilisateurs aux ressources limitées.

Terme associé : **UTXO SET**

UTXO

PROTOCOLE

Sigle de *Unspent Transaction Output*. Un UTXO est une sortie de transaction qui n'a pas encore été dépensée, c'est-à-dire utilisée comme entrée pour une nouvelle transaction. Les UTXOs représentent la fraction de bitcoins que possède un utilisateur et qui sont actuellement disponibles pour être dépensés. Chaque UTXO est associé à un script de sortie spécifique (`scriptPubKey`), qui définit les conditions nécessaires pour dépenser les bitcoins. Les transactions dans Bitcoin consomment ces UTXOs en entrées (inputs) et créent de nouveaux UTXOs en sorties (outputs). Le modèle d'UTXO est fondamental sur Bitcoin, car il permet de vérifier facilement que les transactions n'essaient pas de dépenser des bitcoins qui n'existent pas ou qui ont déjà été dépensés.

Termes associés : **SCRIPTPUBKEY** **UTXO SET**

UTXO SET

PROTOCOLE | FR : ENSEMBLE D'UTXOS

Désigne l'ensemble de tous les UTXOs existants à un moment donné. Autrement dit, c'est une grosse liste de tous les différents morceaux de bitcoins qui attendent d'être dépensés. Si l'on additionne les montants de tous les UTXOs de l'UTXO set, cela nous donne la masse monétaire totale de bitcoins en circulation. Chaque nœud du réseau Bitcoin conserve son propre UTXO set en temps réel. Il l'actualise au fur et à mesure de la confirmation de nouveaux blocs valides, avec les transactions qu'ils incluent, qui consomment certains UTXOs de l'UTXO set, et qui en créent de nouveaux en contrepartie.

Cet UTXO set est conservé par chaque nœud afin de pouvoir vérifier rapidement si les UTXOs dépensés dans les transactions sont bien légitimes. Cela leur permet de détecter et de rejeter les tentatives de doubles dépenses. L'UTXO set est souvent au cœur d'inquiétudes sur la décentralisation de Bitcoin, car sa taille augmente naturellement très rapidement. Puisqu'il faut en conserver une partie en RAM pour pouvoir procéder à la vérification des transactions en temps raisonnable, il est possible que l'UTXO set rende progressivement l'opération d'un nœud complet trop couteuse. L'UTXO set a également un fort impact sur l'IBD (*Initial Block Download*). Au plus on peut mettre une grande part de l'UTXO set en RAM, au plus l'IBD est rapide. Sur Bitcoin Core, l'UTXO set est stocké dans le dossier nommé `/chainstate`.

Termes associés : **ASSUME UTXO** | **UTXO**

V

VALENCY

RGB

Dans le cadre du protocole RGB, une *Valency* désigne un droit public qui ne nécessite pas le stockage explicite d'un état, mais qui peut être ultérieurement rédemptée via une *State Extension*. Ce mécanisme constitue une possibilité préalablement déclarée dans la logique du contrat, accessible à tous ou à certains acteurs spécifiques, et permet de déclencher une extension spécifique de l'état contractuel lorsque les conditions définies sont remplies. La *Valency* offre ainsi une solution flexible et modulable pour anticiper des évolutions futures du contrat.

Termes associés : **CONTRACT STATE** | **STATE EXTENSION**

VALIDATIONLESS MINING

MINAGE | FR : MINAGE SANS VALIDATION

Pratique de minage qui consiste à commencer à travailler sur un nouveau bloc candidat dès la réception du hash de l'en-tête du bloc précédent, sans en avoir téléchargé ni vérifié l'intégralité des transactions. Aussi appelé *SPV mining*, ce procédé permet au mineur de gagner quelques secondes par rapport à ceux qui attendent la validation complète du bloc avant de commencer leur propre travail.

Termes associés : **MINING POOL** | **SPY MINING**

VALUE OVERFLOW INCIDENT

HISTOIRE

Bug exploité le 15 août 2010 sur le réseau Bitcoin, qui a conduit à la création de plus de 184 milliards de bitcoins dans une seule transaction. L'attaquant a exploité une vulnérabilité de dépassement d'entier (*integer overflow*) dans la vérification des montants : deux sorties de 92 233 720 368,54 BTC chacune, proches du maximum représentable par un entier signé sur 64 bits, ont provoqué un débordement lors de l'addition, ce qui a produit un résultat positif inférieur au seuil de détection.

Le problème a été repéré rapidement par Jeff Garzik, qui a alerté la communauté sur le forum BitcoinTalk. Satoshi Nakamoto a publié un correctif dans les heures qui ont suivi. Ce correctif ajoutait une vérification explicite de chaque montant de sortie ainsi que de leur somme, et rejetait toute transaction qui contenait des valeurs négatives ou excessives. Les mineurs ont alors adopté la nouvelle version du logiciel et ont produit une chaîne alternative qui excluait la transaction invalide.

La chaîne corrigée a dépassé la chaîne qui contenait le bug au bloc 74 691, après environ 53 blocs de perturbation. Cet épisode, enregistré sous l'identifiant CVE-2010-5139, constitue le premier incident majeur de l'histoire de Bitcoin et a conduit à la mise en place du système d'alerte du réseau.

Termes associés : **BITCOIN CORE** | **SOFT FORK**

VANITY ADDRESS

PORTEFEUILLE | FR : ADRESSE PERSONNALISÉE

Adresse de réception personnalisée qui contient une séquence spécifique de caractères choisie par l'utilisateur, généralement pour des raisons esthétiques. Ces adresses sont générées en exécutant un processus de calcul, où de multiples clés privées sont créées jusqu'à ce que l'une d'entre elles corresponde à une adresse de réception contenant la séquence désirée. Ce processus ne compromet pas la sécurité de l'adresse, mais peut nécessiter un temps et des ressources de calcul considérables, surtout pour des séquences plus longues ou plus spécifiques. C'est une sorte de processus de *brute force*.

Termes associés : **ADRESSE DE RÉCEPTION** | **VANITYGEN**

VANITYGEN

OUTIL

Premier logiciel open source en ligne de commande utilisé pour créer des adresses de réception personnalisées (*vanity address*). Il fonctionne en générant et en testant par tâtonnement des paires de clés jusqu'à ce qu'une adresse de réception correspondant aux critères spécifiés par l'utilisateur (comme une certaine séquence de caractères spécifiques) soit trouvée. Vanitygen nécessite un processus de calcul intensif, particulièrement pour de longues séquences.

Termes associés : **ADRESSE DE RÉCEPTION** | **VANITY ADDRESS**

VARIANCE

MINAGE

Dans le cadre du minage, désigne les fluctuations des revenus d'un mineur dues à la nature probabiliste de la recherche de blocs valides. Même avec une puissance de calcul constante, le temps nécessaire pour trouver un bloc peut considérablement varier. Par conséquent, les mineurs peuvent passer de longues périodes sans recevoir de récompenses, suivies de périodes de gains importants.

Les pools de minage sont conçus pour réduire cette variance. En mutualisant les puissances de calcul de nombreux hacheurs individuels, les pools augmentent la fréquence à laquelle des blocs sont trouvés, ce qui permet une distribution plus régulière et prévisible des récompenses. On dit ainsi que la variance est réduite.

La variance est un défaut inhérent au mécanisme de la preuve de travail de Bitcoin qui pousse au regroupement du minage.

Termes associés : **LUCK** | **MINING POOL**

VAULT

PORTEFEUILLE | FR : COFFRE-FORT

Mécanisme de sécurité avancé pour la conservation de bitcoins, fondé sur un processus de dépense en deux étapes qui impose un délai entre l'annonce d'un retrait et son exécution effective. Lors de la première transaction, l'utilisateur signale son intention de déplacer les fonds. Une période d'attente s'écoule ensuite, durant laquelle le propriétaire légitime peut détecter toute tentative de vol et rediriger les fonds vers une adresse de récupération sécurisée (« *clawback* »). Si aucune intervention n'a lieu, une seconde transaction finalise le transfert.

Plusieurs approches techniques ont été proposées pour mettre en œuvre des vaults. La première repose sur des transactions présignées : les transactions de dépense sont signées à l'avance, puis les clés de signature sont détruites afin de garantir que seuls les chemins prédéfinis restent utilisables. Cette méthode ne nécessite aucune modification du protocole, mais elle impose une gestion rigoureuse de la destruction des clés. La seconde approche, plus élégante, s'appuie sur les covenants, c'est-à-dire des contraintes appliquées directement dans les scripts pour limiter la manière dont un UTXO peut être dépensé. La proposition `OP_VAULT`, formalisée dans le BIP-0345, introduit des opcodes dédiés (`OP_VAULT` et `OP_VAULT_RECOVER`) qui simplifient considérablement la construction de vaults au niveau du consensus, mais cette proposition a été abandonnée au profit d'`OP_CHECKCONTRACTVERIFY` qui est plus généraliste.

Les vaults sont particulièrement adaptés au stockage à froid institutionnel, où la sécurité prime sur la rapidité d'accès aux fonds.

Termes associés : **COVENANT** | **TIMELOCK**

VBYTES

PROTOCOLE | FR : OCTETS VIRTUELS

Unité de mesure utilisée pour évaluer la taille d'une transaction Bitcoin en tenant compte de la pondération introduite par SegWit. Un vByte (octet virtuel) représente une unité de poids normalisée : 4 unités de poids (WU) correspondent à 1 vByte. Cette mesure permet de comparer équitablement les transactions SegWit et les transactions *legacy*.

Avant SegWit, la taille d'une transaction était mesurée en octets bruts. Avec SegWit, les données de témoin (*witness*) bénéficient d'une réduction : chaque octet de témoin compte pour 1 WU, contre 4 WU pour les autres données. Le calcul en vBytes lisse cette différence. Par exemple, une transaction dont le poids total est de 600 WU fait 150 vBytes.

Les frais de transaction sont calculés en satoshis par vByte (sat/vB). Utiliser les vBytes permet aux portefeuilles d'estimer correctement les frais quelle que soit la structure de la transaction. Un bloc Bitcoin peut contenir au maximum 4 millions de WU, soit l'équivalent d'1 million de vBytes.

Termes associés : **FRAIS DE TRANSACTION** | **SEGWIT**

VERSIONNAGE

‹/› INFORMATIQUE | EN : VERSIONING

Dans le contexte de Bitcoin, désigne le processus d'assignation de versions spécifiques à divers éléments du protocole, afin de faciliter leur suivi, leur gestion et leur mise à jour. Les versions sont numérotées de manière séquentielle, permettant d'identifier les modifications apportées, les nouvelles fonctionnalités ou les corrections de bugs. Par exemple, chaque bloc de la blockchain contient un numéro de version, indiquant les règles du consensus et les structures de données appliquées lors de sa création. On retrouve des versionnages pour les types de scripts, pour les transactions, pour les blocs, pour le logiciel Bitcoin Core, pour les communications réseaux...

Termes associés : **NVERSION** | **SOFT FORK**

VEXL

✱ OUTIL

Application mobile open source qui permet d'acheter et de vendre du bitcoin en pair à pair, sans vérification d'identité (KYC). Lancée en 2022, Vexl est opérée par la Vexl Foundation, un organisme à but non lucratif qui bénéficie du soutien de SatoshiLabs, la société à l'origine de Trezor.

L'application repose sur un principe original : la découverte des contreparties se fait à travers le graphe social de l'utilisateur, c'est-à-dire ses contacts téléphoniques et les contacts de ses contacts. Ce mécanisme reproduit numériquement la confiance des échanges en personne.

Toutes les communications sont chiffrées de bout en bout, aucune donnée personnelle n'est stockée, et chaque profil public est unique à chaque transaction.

Termes associés : **KYC - KNOW YOUR CUSTOMER** | **P2P - PAIR-À-PAIR**

VIABTC

↗ MINAGE

Pool de minage de Bitcoin fondée en 2016 en Chine. ViaBTC compte parmi les principales pools de minage au monde en termes de hashrate. Elle se distingue par la diversité de ses méthodes de rémunération : PPS+, PPLNS et *solo mining*, permettant aux mineurs de choisir le modèle le plus adapté à leur profil de risque.

Termes associés : **MINAGE** | **MINING POOL**

VIN

PROTOCOLE

Élément spécifique d'une transaction Bitcoin qui spécifie la source des fonds utilisés pour satisfaire les outputs. Chaque `vin` fait référence à un output non dépensé (UTXO) d'une transaction précédente. Une transaction peut contenir plusieurs inputs, chacun étant identifié par une combinaison du `txid` (l'identifiant de la transaction d'origine) et du `vout` (l'index de l'output dans cette transaction).

Chaque `vin` inclut les informations suivantes :

- `txid` : l'identifiant de la transaction précédente contenant l'output utilisé ici en input ;
- `vout` : l'index de l'output dans la transaction précédente ;
- `scriptSig` ou `scriptWitness` : un script de déverrouillage qui fournit les données nécessaires pour satisfaire les conditions posées par le `scriptPubKey` de la transaction précédente dont les fonds sont dépensés, généralement en fournissant une signature cryptographique ;
- `nSequence` : un champ spécifique utilisé pour indiquer la manière dont cet input est verrouillé dans le temps, ainsi que d'autres options comme RBF.

Terme associé : **NSEQUENCE**

VIRGIN BITCOIN

CONFIDENTIALITÉ | FR : BITCOIN VIERGE

Unité de bitcoin fraîchement émise dans une transaction coinbase (récompense de bloc) et qui n'a encore jamais été dépensée dans une transaction ultérieure. Son historique de transactions est donc vierge de toute association avec des activités identifiées par les outils d'analyse de chaîne.

Termes associés : **ANALYSE DE CHAINE** | **COINBASE**

VLS

LIGHTNING NETWORK

Sigle de « Validating Lightning Signer ». Bibliothèque open source en Rust qui sépare les clés privées d'un nœud Lightning du logiciel du nœud lui-même, en les déportant vers un dispositif de signature dédié. Avant chaque opération cryptographique, VLS applique un ensemble de plus de cinquante règles de validation pour vérifier que la transaction proposée est sûre.

Si le nœud est compromis ou tente une action malveillante, comme signer une *commitment transaction* révoquée ou fermer un canal vers une adresse non autorisée, VLS refuse la signature. Ce mécanisme protège les fonds même en cas de compromission totale du logiciel du nœud.

VLS propose des intégrations de référence pour CLN et LDK. Le projet est encore en phase de développement actif et n'a pas atteint le stade de version de production stable.

Termes associés : **LDK - LIGHTNING DEV KIT** | **LIGHTNING NETWORK**

VOLATILITÉ

$ ÉCONOMIE ET RÉGULATION | EN : VOLATILITY

Mesure de l'amplitude des variations du prix d'un actif sur une période donnée. Dans le contexte de Bitcoin, la volatilité désigne les fluctuations parfois importantes du cours du BTC par rapport aux monnaies fiat comme le dollar ou l'euro. Une volatilité élevée signifie que le prix peut connaître de fortes hausses ou baisses en peu de temps.

VOLTAGE

ORGANISATION

Plateforme d'infrastructure cloud dédiée au Lightning Network. Voltage permet aux entreprises et aux développeurs de déployer et de gérer des nœuds Lightning (LND, Core Lightning) sans avoir à administrer l'infrastructure sous-jacente.

La plateforme propose l'hébergement de nœuds, des outils de surveillance et des services de gestion de liquidité. Ces fonctionnalités facilitent l'intégration du Lightning Network dans des applications commerciales.

Le modèle de Voltage est non-custodial : l'opérateur du nœud conserve le contrôle de ses clés et de ses fonds.

Termes associés : **LIGHTNING NETWORK** | **NOEUD LIGHTNING**

VOUT

PROTOCOLE

Élément spécifique d'une transaction Bitcoin qui détermine la destination des fonds envoyés. Une transaction peut inclure plusieurs outputs, chacun étant distinctement identifié par la combinaison de l'identifiant de la transaction (`txid`) et d'un index appelé `vout`. Cet index, qui commence à `0`, marque la position de l'output dans la séquence des outputs de la transaction. Ainsi, le premier output sera désigné par `"vout": 0`, le second par `"vout": 1`, et ainsi de suite.

Chaque `vout` encapsule principalement deux informations :

- la valeur, exprimée en bitcoins, qui représente le montant envoyé ;
- un script de verrouillage (`scriptPubKey`) qui stipule les conditions requises pour que les fonds puissent être dépensés de nouveau dans une prochaine transaction.

La combinaison du `txid` et du `vout` d'une pièce spécifique forme ce que l'on appelle un *outpoint*, par exemple :

- txid :

```
4c160086e39a940c2459f03bb7cfe5b768fc78373c9960dc2cf2fa61b57d0adf
```

- vout :

```
0
```

- outpoint :

```
4c160086e39a940c2459f03bb7cfe5b768fc78373c9960dc2cf2fa61b57d0adf
:0
```

Termes associés : **OUTPUT** | **UTXO**

VPRV

PORTEFEUILLE

Préfixe de clé privée étendue pour les comptes SegWit V0 sur Bitcoin Testnet.

Terme associé : **CLÉ ÉTENDUE**

VPUB

PORTEFEUILLE

Préfixe de clé publique étendue pour les comptes SegWit V0 sur Bitcoin Testnet.

Terme associé : **CLÉ ÉTENDUE**

VTXO

COUCHE SUPÉRIEURE

Sigle de « *Virtual Transaction Output* ». Un VTXO est une part virtuelle d'un UTXO Bitcoin on-chain au sein du protocole Ark. Plutôt que de posséder chacun un UTXO individuel sur la blockchain (coûteux en frais et difficilement scalable), plusieurs utilisateurs partagent un même UTXO via des VTXO gérés par un *Ark Service Provider* (ASP).

Un VTXO est obtenu soit en déposant du bitcoin on-chain vers une adresse d'embarquement (*boarding address*), soit en le recevant d'un autre utilisateur du même ASP. Les VTXO sont connectés atomiquement entre eux : lorsqu'Alice envoie un VTXO à Bob, l'ASP crée une sortie on-chain incluant la part d'Alice destinée à Bob. Alice ne perd ses fonds que si Bob les reçoit effectivement. Les VTXO existants sont détruits et recréés en continu lors des rounds organisés par l'ASP.

Les VTXO possèdent une date d'expiration (typiquement 4 semaines). L'utilisateur doit les rafraîchir avant l'échéance sous peine que l'ASP récupère la liquidité. À tout moment, l'utilisateur peut effectuer une sortie unilatérale on-chain pour convertir son VTXO en UTXO classique, bien que cette opération implique des frais de transaction.

Termes associés : **ARK** | **UTXO**

VULNERABILITE DE MURCH-ZAWY

PROTOCOLE | EN : MURCH-ZAWY VULNERABILITY

Attaque théorique sur le mécanisme d'ajustement de la difficulté de Bitcoin, décrite par Zawy et formalisée par Murch (Mark Erhardt) en août 2024. Contrairement à l'attaque *time warp* classique, cette vulnérabilité n'exploite pas l'erreur *off-by-one* dans le calcul de la durée des périodes de retarget, mais repose sur l'alternance stratégique des horodatages au dernier bloc de chaque période de difficulté.

L'attaquant, qui contrôle une majorité de la puissance de calcul, maintient les horodatages au minimum autorisé par la règle du *Median Time Past* (MTP)

durant chaque période, puis fixe l'horodatage du dernier bloc loin dans le futur pour réduire la difficulté d'un facteur 4. En alternant ensuite entre un horodatage minimal et un horodatage futur pour les derniers blocs de chaque période, il fait osciller la difficulté entre un quart et un seizième de la valeur initiale. Certaines périodes présentent alors une durée calculée négative, mais le protocole la traite comme une demi-semaine, ce qui ne relève la difficulté que d'un facteur 4.

En pratique, l'attaque nécessite environ 16 semaines de minage égoïste. À l'issue de cette période, l'attaquant publie sa chaîne secrète, réorganise environ 8 064 blocs et collecte les récompenses d'environ 39 816 blocs. Murch a proposé une variante qui réduit la difficulté de façon exponentielle en augmentant l'horodatage futur par paliers.

Le BIP-0054 (Consensus Cleanup) corrige cette vulnérabilité en imposant que l'horodatage du dernier bloc d'une période de difficulté soit supérieur ou égal à celui du premier bloc de la même période.

Termes associés : **CONSENSUS CLEANUP** | **SELFISH MINING** | **TIME WARP**

W

WABISABI

CONFIDENTIALITÉ

Protocole de coinjoins à coordinateur central utilisé sur le portefeuille Wasabi 2.0. Ce modèle a été conçu par Ádám Ficsór (nopara73), Yuval Kogman, Lucas Ontivero et István András Seres en 2021, et a été intégré dans le logiciel Wasabi 2.0 l'année suivante.

Terme associé : **CHAUMIAN COINJOIN**

WALLET.DAT

PROTOCOLE

Fichier dans Bitcoin Core qui stocke des informations sur le portefeuille de l'utilisateur, telles que les clés privées et les transactions effectuées. Le fichier `wallet.dat` peut être chiffré par l'utilisateur pour assurer la sécurité des fonds, mais il ne l'est pas par défaut. Depuis la version 0.16.0, ce fichier a été déplacé dans le dossier `/wallets`.

Termes associés : **BITCOIN CORE** **CLÉ PRIVÉE**

WIF - WALLET IMPORT FORMAT

PORTEFEUILLE

Méthode pour encoder une clé privée Bitcoin de manière à ce qu'elle puisse être importée ou exportée plus facilement entre différents portefeuilles. Le WIF est établi sur un encodage `Base58Check`, qui inclut des informations sur la version, la compression de la clé publique correspondante et une somme de contrôle pour la détection d'erreurs de saisie. Une clé privée WIF commence par les caractères `5` pour les clés non compressées, ou `K` et `L` pour les clés compressées, et contient tous les caractères nécessaires pour reconstituer la clé privée originale. Le format WIF fournit un moyen standardisé pour transférer une clé privée entre différents logiciels de portefeuille.

Termes associés : **BASE58CHECK** **CLÉ PRIVÉE**

WALLET OF SATOSHI

LIGHTNING NETWORK

Portefeuille mobile spécialisé dans les paiements via le Lightning Network. Wallet of Satoshi est conçu pour offrir une expérience d'utilisation simplifiée : l'utilisateur peut envoyer et recevoir des paiements Lightning en quelques secondes, sans aucune configuration technique.

Historiquement, Wallet of Satoshi fonctionnait sur un modèle *custodial* : l'entreprise détenait les clés privées et gérait les fonds au nom de l'utilisateur. Ce modèle simplifie l'utilisation, mais implique un compromis : l'utilisateur doit faire confiance au fournisseur pour la garde de ses bitcoins. Depuis 2025, WoS propose également en option une intégration du protocole Spark, permettant ainsi d'adopter un modèle *self-custodial*.

Termes associés : **LIGHTNING NETWORK** **LNURL**

WALLETS/DB.LOG

PROTOCOLE

Fichier journal dans Bitcoin Core spécifique à la base de données des portefeuilles. Il enregistre les opérations et les événements liés à la base de données des portefeuilles pour la résolution de problèmes.

Termes associés : **BITCOIN CORE** | **WALLET.DAT**

WASABI WALLET

PORTEFEUILLE

Portefeuille Bitcoin axé sur la confidentialité offrant des fonctionnalités telles que le coinjoin.

Termes associés : **COINJOIN** | **PORTEFEUILLE**

WATCH-ONLY WALLET

PORTEFEUILLE | FR : PORTEFEUILLE EN LECTURE SEULE

Un watch-only wallet est un type de logiciel qui permet à un utilisateur de voir les transactions associées à une clé ou un ensemble de clés Bitcoin spécifiques, sans posséder les clés privées correspondantes. Il offre une visibilité sur le solde et l'historique des transactions, sans pour autant permettre de dépenser les fonds du portefeuille. Par exemple, l'application Sentinel est un watch-only wallet.

Termes associés : **CLÉ PRIVÉE** | **XPUB**

WATCHMEN

SIDECHAIN | FR : GARDIENS

Dans le cadre de Liquid (une sidechain de Bitcoin), ce sont des entités chargées de maintenir l'ancrage du L-BTC, le jeton natif de Liquid, en gérant et sécurisant les BTC utilisés en sous-jacent. Ils s'assurent que les actifs transférés entre la blockchain Bitcoin principale et la sidechain Liquid sont correctement verrouillés et déverrouillés. L'objectif de leurs actions est de maintenir la même valeur entre le L-BTC circulant sur Liquid et le BTC circulant sur Bitcoin. Dans Liquid, les watchmen font partie des fonctionnaires avec les *blocksigners*.

Termes associés : **BLOCKSIGNERS** | **LIQUID NETWORK**

WATCHTOWER

LIGHTNING NETWORK | FR : TOUR DE GARDE

Service externe qui surveille la blockchain Bitcoin à la place d'un nœud Lightning pour détecter toute tentative de triche sur ses canaux. Si un pair malhonnête publie une ancienne transaction d'engagement pendant que le nœud est hors ligne, la watchtower diffuse automatiquement la transaction de justice correspondante pour récupérer l'intégralité des fonds du canal avant l'expiration du *timelock*.

Le mécanisme est conçu pour préserver la confidentialité du nœud : à chaque mise à jour d'un canal, le nœud chiffre la transaction de pénalité pré-signée avec le TXID de la transaction d'engagement concernée, puis transmet à la watchtower ce blob chiffré accompagné d'une portion tronquée du TXID. La watchtower ne peut déchiffrer la transaction que si elle observe un TXID correspondant sur la blockchain grâce à la portion tronquée, c'est-à-dire en cas de triche effective. Elle n'a donc aucune visibilité sur l'identité des pairs, les soldes ou la structure des canaux en fonctionnement normal. Elle apprendra ces informations uniquement en cas de triche.

La watchtower constitue un tiers de confiance minimal : elle ne peut ni modifier ni détourner les fonds, puisque la transaction de justice est déjà signée avec des sorties figées en faveur du propriétaire du canal. Le seul risque pour le nœud consiste à s'appuyer sur une watchtower qui ne remplirait pas son rôle au moment où elle est réellement nécessaire. C'est pourquoi un nœud peut en configurer plusieurs, afin de s'assurer qu'au moins l'une d'entre elles agira correctement en cas de besoin.

Il existe des watchtowers dites « altruistes », qui sont proposées publiquement et gratuitement, et n'importe quel nœud Lightning peut également jouer ce rôle pour d'autres nœuds.

Termes associés : **FORCE CLOSE** | **TRANSACTION D'ENGAGEMENT**

WEIGHT UNIT - WU

PROTOCOLE | FR : UNITÉ DE POIDS

Unité de mesure introduite par SegWit (BIP-0141) pour quantifier l'espace qu'une transaction ou un bloc occupe par rapport à la limite maximale autorisée. Chaque *weight unit* (WU) représente 1/4 000 000e de la capacité maximale d'un bloc. Cette unité remplace la mesure en octets bruts qui était utilisée avant SegWit, et permet d'appliquer une pondération différenciée selon la nature des données dans une transaction.

Le calcul du poids d'une transaction repose sur une distinction entre les données *non-witness* (version, inputs, outputs, *locktime*) et les données *witness* (signatures, clés publiques et autres éléments de preuve ajoutés par SegWit). La formule est la suivante :

$$poids \;=\; taille\ sans\ witness \times 4 \;+\; taille\ du\ witness \times 1$$

Autrement dit, chaque octet de données *non-witness* compte pour 4 WU, tandis

que chaque octet de données *witness* ne compte que pour 1 WU. Pour une transaction *legacy* (sans SegWit), il n'y a pas de données *witness*, donc le poids est simplement 4 fois la taille en octets. Par exemple, une transaction *legacy* P2PKH classique de 226 octets pèse $226 \times 4 = 904$ WU.

Pour une transaction SegWit équivalente (P2WPKH, 1 input, 2 outputs), la taille totale est d'environ 222 octets, dont environ 110 octets de *witness* (signature et clé publique). Le poids est alors :

$$112 \times 4 + 110 \times 1 = 448 + 110 = 558\ WU$$

Cela représente une réduction d'environ 38 % par rapport à la transaction *legacy* équivalente, ce qui permet d'inclure davantage de transactions dans un bloc.

La limite de bloc est fixée à 4 000 000 WU. Si un bloc ne contient que des transactions *legacy*, cette limite est équivalente à l'ancienne limite de 1 000 000 d'octets ($1\,000\,000 \times 4 = 4\,000\,000$), ce qui assure la rétrocompatibilité avec les nœuds non mis à jour. En pratique, avec une majorité de transactions SegWit, les blocs atteignent environ 2 à 3 Mo en taille réelle. Le maximum théorique approche 4 Mo, mais uniquement dans le cas pathologique d'un bloc rempli presque exclusivement de données *witness*.

Pour convertir un poids en octets virtuels (*vbytes*), il suffit de diviser par 4 : $vbytes = poids\ /\ 4$. Les frais de transaction sont généralement exprimés en sat/vB (satoshis par *vbyte*).

Termes associés : MAX_BLOC_SIZE | SEGWIT | VBYTES

WHALE

ÉCONOMIE ET RÉGULATION | FR : BALEINE

Désigne un individu ou une entité qui possède une quantité très importante de bitcoins.

Terme associé : HODL

WHIRLPOOL

CONFIDENTIALITÉ

Protocole de coinjoins chaumiens ZeroLink, développée par les équipes du portefeuille Samourai Wallet. Whirlpool était disponible sur les portefeuilles Samourai Wallet (Android), Sparrow Wallet (PC) et Bitcoin Keeper (IOS et Android).

Termes associés : CHAUMIAN COINJOIN | ZEROLINK

WHIRLPOOL STAT TOOL

CONFIDENTIALITÉ

Logiciel en ligne de commandes développé par Samourai Wallet qui permet de fournir les anonsets prospectifs et rétrospectifs d'une pièce mixée au sein de Whirlpool, ainsi que son taux de diffusion dans la pool. WST utilise l'algorithme HyperLogLogPlusPlus qui permet d'estimer le nombre de valeurs distinctes dans un très grand ensemble de données.

Termes associés : SAMOURAI WALLET | WHIRLPOOL

WHITE PAPER

HISTOIRE | FR : LIVRE BLANC

Rapport ou guide qui informe les lecteurs de manière concise sur une question complexe. Dans le cas de Bitcoin, il s'agit du nom donné au document scientifique nommé « *Bitcoin : A Peer-to-Peer Electronic Cash System* » publié par Satoshi Nakamoto pour la première fois le 31 octobre 2008. Ce document décrit les principes fondamentaux de Bitcoin. Le White Paper que l'on présente de nos jours est une seconde version publiée le 24 mars 2009.

Pour télécharger le White Paper de Bitcoin : https://bitcoin.org/bitcoin.pdf.

Termes associés : CYPHERPUNKS | NAKAMOTO SATOSHI

WITNESS TRANSACTION

RGB

Dans le cadre du protocole RGB, la Witness Transaction désigne la transaction Bitcoin qui clôt le Single-use Seal autour d'un message intégrant un Multi Protocol Commitment (MPC). Cette opération consiste à dépenser un UTXO existant, fermant ainsi le sceau précédent, tout en créant un nouvel output contenant un engagement vers une nouvelle définition de sceau, afin de verrouiller l'engagement contractuel inscrit dans le protocole. La Witness Transaction constitue donc une preuve on-chain que l'état du contrat RGB a été fixé à un instant précis.

Termes associés : MULTI PROTOCOL COMMITMENT | SINGLE-USE SEAL

WITNESSSCRIPT

SCRIPT

Script qui spécifie les conditions sous lesquelles les bitcoins peuvent être dépensés dans les UTXOs P2WSH ou P2SH-P2WSH. Typiquement, les `witnessScript` déterminent les conditions d'un portefeuille multisignatures sous standard SegWit. Dans ces standards de script, le `scriptPubKey` de l'UTXO (la sortie) contient un hachage du `witnessScript`. Pour utiliser cet UTXO comme entrée dans une nouvelle transaction, le détenteur doit révéler le `witnessScript` original, afin de prouver sa correspondance avec l'empreinte dans le `scriptPubKey`. Le `witnessScript` doit alors être inclus dans le `scriptWitness` de la transaction, qui contient également les éléments nécessaires pour valider le script, comme par exemple les signatures. Le `witnessScript` est donc l'équivalent pour SegWit du `redeemScript` dans une transaction P2SH, à la différence près qu'il est placé dans le témoin de la

transaction, et non dans le `scriptSig`.

Attention, le `witnessScript` ne doit pas être confondu avec le `scriptWitness`. Tandis que le `witnessScript` définit les conditions de dépense d'un UTXO P2WSH ou P2SH-P2WSH et constitue un script à part entière, le `scriptWitness` contient les données de témoin de tout input SegWit.

Termes associés : **P2WSH** | **REDEEMSCRIPT**

WIZARDSARDINE

COMMUNAUTÉ

Entreprise spécialisée dans la sécurité et la conservation de bitcoins, fondée en 2020 par Kevin Loaec et Antoine Poinsot, et incorporée aux Açores (Portugal). Wizardsardine développe des solutions open source de gestion de portefeuilles Bitcoin, avec un accent sur les mécanismes de sécurité avancés comme les *vaults* et les politiques de dépense configurables.

L'entreprise est à l'origine du protocole Revault, un système de conservation institutionnel utilisant des transactions présignées et des *watchtowers* pour sécuriser les fonds d'organisations. Wizardsardine développe également Liana, un portefeuille Bitcoin qui exploite Miniscript pour offrir des configurations de récupération et d'héritage basées sur des *timelocks*.

Termes associés : **LIANA** | **REVAULT**

WRENCH ATTACK

ATTAQUE | FR : ATTAQUE À LA CLÉ À MOLETTE

Attaque physique qui vise à contraindre un détenteur de bitcoins à céder ses fonds sous la menace ou par la violence. Le nom fait référence à une clé à molette bon marché (d'une valeur symbolique de 5 dollars) utilisée comme outil de coercition, et illustre l'idée que la cryptographie la plus robuste peut être contournée par une simple agression physique.

Plusieurs contre-mesures existent pour limiter les risques liés à cette attaque. L'utilisation d'une passphrase BIP-0039 permet de créer un portefeuille caché, ce qui offre une forme de déni plausible. Le recours au multisig avec une distribution géographique des clés empêche un attaquant d'obtenir l'ensemble des signatures nécessaires en un seul lieu. Les timelocks peuvent aussi être employés pour rendre les fonds temporairement inaccessibles. Enfin, la discrétion sur ses avoirs en bitcoins reste la mesure préventive la plus élémentaire.

Cette notion rappelle que la sécurité dans l'écosystème Bitcoin est un enjeu global qui dépasse la seule dimension informatique. Elle implique également des considérations de sécurité opérationnelle (OPSEC) et de protection physique.

Termes associés : **CLÉ PRIVÉE** | **MULTISIG** | **PASSPHRASE - BIP-0039**

WTXID

PROTOCOLE

Extension du TXID traditionnel, incluant les données de témoin (*witness*) introduites avec SegWit. Alors que le TXID est un hachage des données de transaction hors témoin, le WTXID est le `SHA256d` de l'intégralité des données de la transaction, témoin inclus. Les WTXID sont stockés dans un second arbre de Merkle dont la racine est mise dans la transaction coinbase. Cette séparation permet de supprimer la malléabilité du TXID de la transaction.

Termes associés : **SEGWIT** | **TXID - TRANSACTION IDENTIFIER**

WUMBO CHANNEL

LIGHTNING NETWORK

Canal Lightning de grande capacité, dépassant la limite historique de 16 777 215 satoshis (environ 0,168 BTC) imposée par défaut aux premiers nœuds. Cette limite, codée sur 24 bits, avait été instaurée comme mesure de prudence durant les premières années du Lightning Network pour limiter les pertes potentielles en cas de bug dans les implémentations.

Par la suite, les implémentations ont introduit une option permettant d'accepter des canaux allant jusqu'à 10 BTC. Ces canaux à grande capacité sont baptisés « Wumbo channels », en référence à un épisode de *Bob l'éponge*.

Termes associés : **CANAL DE PAIMENT** | **LIGHTNING NETWORK**

X

X-ONLY PUBLIC KEYS

CRYPTOGRAPHIE | FR : CLÉS PUBLIQUES X-ONLY

Format de clé publique sur la courbe elliptique secp256k1 dans lequel seule la coordonnée x du point est conservée, sans l'octet de préfixe qui indique habituellement la parité de la coordonnée y. Ce format réduit la taille d'une clé publique de 33 octets (clé compressée classique) à 32 octets.

Sur une courbe elliptique, chaque valeur de x correspond à exactement deux points symétriques par rapport à l'axe des abscisses, qui ne diffèrent que par le signe de leur coordonnée y. Pour lever cette ambiguïté sans recourir à un octet supplémentaire, le BIP-0340 (signatures de Schnorr) impose une convention : seul le point dont la coordonnée y est paire est considéré comme valide. Ainsi, le vérificateur peut reconstituer le point complet à partir des seuls 32 octets de x.

Ce format est utilisé de façon systématique dans Taproot (BIP-0341) et Tapscript (BIP-0342) pour les clés internes, les clés de sortie et les vérifications de signature au sein des scripts SegWit v1. L'économie d'un octet par clé, bien que modeste en apparence, se cumule sur l'ensemble des transactions du réseau et contribue à réduire la taille des témoins.

Le recours aux clés x-only introduit cependant une contrainte pour les protocoles d'agrégation de clés comme MuSig2 : les participants doivent s'assurer que la clé agrégée finale produit bien un point à coordonnée y paire, ce qui peut nécessiter une étape de négation de la clé privée.

Termes associés : **SCHNORR** | **TAPROOT**

XBT

ÉCONOMIE ET RÉGULATION

Symbole boursier et monétaire (*ticker*) utilisé pour représenter le bitcoin, conforme à la norme ISO 4217 qui régit les codes monétaires. Contrairement à « BTC », qui est le symbole historique, « XBT » respecte la convention selon laquelle les monnaies non adossées à un État commencent par un « X ». Ainsi, bien que non officiellement reconnu par l'ISO, « XBT » est utilisé par certaines organisations pour représenter les unités de bitcoin.

Terme associé : **BTC**

XMR - MONERO

ÉCONOMIE ET RÉGULATION

Cryptomonnaie alternative (altcoin) lancée en avril 2014, axée sur la confidentialité des transactions. Monero (XMR) utilise plusieurs techniques cryptographiques pour dissimuler par défaut l'identité des expéditeurs, des destinataires et les montants échangés : les signatures de cercle (*ring signatures*), les adresses furtives (*stealth addresses*) et le protocole RingCT (*Ring Confidential Transactions*). Contrairement à Bitcoin, où les transactions sont pseudonymes et traçables sur une blockchain publique, Monero rend l'analyse de chaîne nettement plus difficile. Son algorithme de minage, RandomX, est conçu pour résister aux ASIC et favoriser le minage par processeur (CPU).

Termes associés : **ALTCOIN** **COINJOIN**

XOR

CRYPTOGRAPHIE | FR : OU EXCLUSIF

Sigle de l'opération « Exclusive or ». C'est une fonction fondamentale de la logique booléenne. Cette opération prend deux opérandes booléens, chacun étant $vrai$ ou $faux$, et produit une sortie $vraie$ uniquement lorsque les deux opérandes diffèrent. Autrement dit, la sortie de l'opération XOR est $vraie$ si exactement un (mais pas les deux) des opérandes est $vrai$. Par exemple, l'opération XOR entre 1 et 0 donnera comme résultat 1. Nous noterons :

$$1 \oplus 0 = 1$$

De même, l'opération XOR peut être effectuée sur des séquences plus longues de bits. Par exemple :

$$10110 \oplus 01110 = 11000$$

Chaque bit de la séquence est comparé à son homologue et l'opération XOR est appliquée. Voici la table de vérité de l'opération XOR :

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

L'opération XOR est utilisée dans de nombreux domaines de l'informatique, notamment dans la cryptographie, pour ses attributs intéressants comme :

- Sa commutativité : l'ordre des opérandes n'affecte pas le résultat. Pour deux variables D et E données : $D \oplus E = E \oplus D$;
- Son associativité : le regroupement des opérandes n'affecte pas le résultat. Pour trois variables A, B et C données : $(A \oplus B) \oplus C = A \oplus (B \oplus C)$;
- Il dispose d'un élément neutre 0 : un opérande xoré à 0 sera toujours égal à l'opérande. Pour une variable A donnée : $A \oplus 0 = A$;
- Chaque élément est son propre inverse. Pour une variable A donnée : $A \oplus$

$A = 0$.

Dans le cadre de Bitcoin, on utilise évidemment l'opération XOR à de nombreux endroits. Par exemple, le XOR est massivement utilisé dans la fonction SHA256, elle-même largement utilisée dans le protocole Bitcoin. Certains protocoles comme le *SeedXOR* de Coldcard utilisent également cette primitive pour d'autres applications. On le retrouve aussi dans le BIP-0047 pour chiffrer le code de paiement réutilisable lors de sa transmission.

Dans le domaine plus général de la cryptographie, le XOR peut être utilisé tel quel comme un algorithme de chiffrement symétrique. On appelle cet algorithme le « *Masque Jetable* » ou le « *Chiffre Vernam* » du nom de son inventeur. Cet algorithme, bien qu'inutile en pratique du fait de la longueur de la clé, est un des seuls algorithmes de chiffrement reconnus comme inconditionnellement sûrs.

Terme associé : **SHA256**

XPRV

PORTEFEUILLE

Préfixe de clé privée étendue pour les comptes Legacy et SegWit V1 sur Bitcoin.

Terme associé : **CLÉ ÉTENDUE**

XPUB

PORTEFEUILLE

Préfixe de clé publique étendue pour les comptes Legacy et SegWit V1 sur Bitcoin.

Terme associé : **CLÉ ÉTENDUE**

Y

YPRV

PORTEFEUILLE

Préfixe de clé privée étendue pour les comptes Nested SegWit sur Bitcoin.

Terme associé : **CLÉ ÉTENDUE**

YPUB

PORTEFEUILLE

Préfixe de clé publique étendue pour les comptes Nested SegWit sur Bitcoin.

Terme associé : **CLÉ ÉTENDUE**

Z

ZAP

LIGHTNING NETWORK

Micro-paiement en bitcoin effectué via le Lightning Network et associé cryptographiquement à un événement Nostr. Défini par le NIP-57, le mécanisme de zap permet à un utilisateur de récompenser directement l'auteur d'une publication en envoyant des satoshis depuis son portefeuille Lightning. Les zaps constituent l'une des intégrations les plus abouties entre Bitcoin et Nostr, permettant un modèle de rémunération direct, sans intermédiaire et sans censure possible.

Termes associés : **LIGHTNING NETWORK** | **LNURL**

ZBD

ORGANISATION

Entreprise fondée en 2019 spécialisée dans l'intégration de paiements Lightning dans l'industrie du jeu vidéo. Anciennement connue sous le nom de Zebedee, elle fournit des API et des SDK qui permettent aux développeurs de jeux d'intégrer des transactions en sats et qui offrent aux joueurs la possibilité de gagner et de dépenser de vrais bitcoins dans des univers virtuels. ZBD propose également une application mobile qui fait office de portefeuille Lightning dédié au gaming, ainsi qu'une infrastructure de paiements en streaming.

Termes associés : **BLOCKSTREAM** | **LIGHTNING NETWORK**

ZEROCONF

PROTOCOLE

Pratique risquée consistant à considérer une transaction Bitcoin comme définitive, et à procéder à l'exécution de l'acte associé en contrepartie (tel que la vente d'un bien ou d'un service), avant que la transaction ne soit réellement incluse dans un bloc sur la blockchain. Les transactions non confirmées, ou en zeroconf, sont vulnérables à des attaques de double dépense, car elles ne sont pas encore irrévocablement inscrites dans le registre. Le zeroconf peut éventuellement être envisagé dans des contextes très spécifiques, où la rapidité est prioritaire, comme dans le cas de petites transactions commerciales ou dans le cas d'une transaction entre proches. Dans ces situations, le risque de double dépense est souvent considéré comme acceptable en comparaison de l'avantage d'une transaction rapide. Néanmoins, pour des transactions importantes, en particulier lorsqu'on ne connaît pas l'expéditeur, il est crucial d'attendre plusieurs confirmations avant de considérer la transaction comme immuable. La norme généralement acceptée est d'attendre 6 confirmations, ce qui signifie que 5 blocs supplémentaires doivent être minés après celui incluant la transaction, pour la considérer comme définitive.

Termes associés : **CONFIRMATION** | **MEMPOOL**

ZEROLINK

CONFIDENTIALITÉ

Protocole de Chaumian coinjoin qui vise à briser toutes les liaisons entre des ensembles de pièces séparées à travers des techniques avancées de mixage. Le protocole ZeroLink se distingue par sa capacité à protéger l'anonymat des utilisateurs contre diverses formes d'analyses de chaîne au niveau de la transaction et du réseau. ZeroLink introduit un cadre pour les portefeuilles de coinjoin, avec l'utilisation de comptes pré-mix et post-mix ségrégués, ainsi qu'une technique de mixage propre : le Chaumian coinjoin. Les deux principales implémentations de ZeroLink ont été Wasabi Wallet (version 1.0) et Whirlpool, disponible sur Samourai Wallet et Sparrow Wallet. Ce protocole a été introduit par nopara73 et TDevD en 2017.

Terme associé : **CHAUMIAN COINJOIN**

ZEROSYNC

PROTOCOLE

Projet développé pour tirer parti des ZKP (preuves à divulgation nulle de connaissance) dans l'écosystème Bitcoin, afin d'améliorer le passage à l'échelle et la confidentialité du système. Leur protocole principal (également nommé *ZeroSync*) permet de compresser l'historique de la blockchain Bitcoin, afin d'accélérer la synchronisation initiale des nœuds avec le réseau via l'utilisation d'une preuve compacte, sans pour autant compromettre la vérification.

Termes associés : **UTREEXO** **ZKP - ZERO-KNOWLEDGE PROOF**

ZEUS

LIGHTNING NETWORK

Application mobile open source qui permet de gérer un nœud Lightning Network depuis un téléphone. Zeus peut fonctionner selon deux modes : la connexion à distance à un nœud Lightning existant (LND, Core Lightning ou Eclair), ou l'utilisation d'un nœud Lightning intégré directement dans l'application grâce à une implémentation LND embarquée. Zeus permet d'effectuer et de recevoir des paiements Lightning et on-chain, de gérer ses canaux de paiement, de consulter l'état de son nœud et de configurer les frais de routage.

Termes associés : **LIGHTNING NETWORK** **LND**

ZKP - ZERO-KNOWLEDGE PROOF

CRYPTOGRAPHIE | FR : PREUVE À DIVULGATION NULLE DE CONNAISSANCE

Méthode cryptographique permettant à une partie (le prouveur) de prouver à une autre partie (le vérificateur) qu'une information est vraie, sans révéler l'information ni aucun aspect de celle-ci. Une ZKP permet de garantir l'exactitude d'une affirmation, tout en préservant la confidentialité des données sous-jacentes.

Terme associé : **ZEROSYNC**

ZKSNACKS

ORGANISATION

Société éditrice de Wasabi Wallet, un portefeuille Bitcoin desktop open source axé sur la confidentialité. zkSNACKs a développé le protocole WabiSabi, un mécanisme de coinjoin utilisé dans Wasabi Wallet 2.0 pour améliorer la confidentialité des transactions Bitcoin. En juin 2024, zkSNACKs a mis fin à son service de coordination coinjoin en raison de l'incertitude réglementaire. Wasabi Wallet continue de fonctionner en tant que portefeuille Bitcoin standard, et d'autres coordinateurs de coinjoin WabiSabi sont apparus.

Termes associés : **COINJOIN** | **WASABI WALLET**

ZMQ - ZEROMQ

INFORMATIQUE

Bibliothèque de messagerie asynchrone utilisée par Bitcoin Core pour notifier en temps réel d'autres logiciels de l'arrivée de nouveaux blocs et de nouvelles transactions. ZMQ fonctionne selon un modèle publication-abonnement (*pub/sub*) : Bitcoin Core publie des événements sur un socket, et les logiciels abonnés (comme LND) les reçoivent instantanément sans avoir à interroger périodiquement le nœud. Cette interface est essentielle pour les implémentations Lightning qui doivent réagir rapidement aux confirmations de transactions, notamment pour la détection des fermetures de canaux et la publication des transactions de justice.

Termes associés : **BITCOIN CORE** | **LND**

ZOMBIE CHANNEL

LIGHTNING NETWORK | FR : CANAL ZOMBIE

Canal Lightning considéré comme inactif, généralement parce que le pair distant a subi une défaillance ou n'est plus en ligne depuis une période prolongée. Un *zombie channel* immobilise du capital sans pouvoir être utilisé pour router des paiements, ce qui réduit la liquidité disponible de l'opérateur de nœud. Ce type de canal peut également être problématique en cas de tentative de récupération d'un nœud Lightning via un SCB (*Static Channel Backup*). Il est recommandé de fermer ces canaux lorsqu'il devient improbable que le pair revienne en ligne.

Terme associé : **CANAL DE PAIMENT**

ZPRV

PORTEFEUILLE

Préfixe de clé privée étendue pour les comptes SegWit V0 sur Bitcoin.

Terme associé : **CLÉ ÉTENDUE**

ZPUB

PORTEFEUILLE

Préfixe de clé publique étendue pour les comptes SegWit V0 sur Bitcoin.

Terme associé : **CLÉ ÉTENDUE**

© 2026 Loïc Morel

Dictionnaire de Bitcoin : Tout le vocabulaire technique de Bitcoin

Version du 23 mars 2026

https://github.com/LoicPandul/Dictionnaire-de-Bitcoin/

Cet ouvrage est sous licence CC BY-NC-SA 4.0

https://creativecommons.org/licenses/by-nc-sa/4.0/

Lightning : sats@pandul.fr

Email : loic@pandul.fr

Site web : https://pandul.fr/

GitHub : https://github.com/LoicPandul/

ISBN : 9798250570664