

INDEX

Note: Page numbers in *italics* refer to comic pages

Numbers & Symbols

µBTC (microbitcoins), 9
51 percent attacks, 167

A

addEventListener function, 234
addition, and elliptic curves, 147–148
AddressFormatException exception
 type, 239
addWallet() function, 240
Adleman, Leonard, 133–134
alternative coins (altcoins), 64
 comic on, *181–184*
Andresen, Gavin, 113–114
anonymity, Bitcoin ATMs and, 62
anonymity by default, 124
anonymous rating service, 208
application specific integrated
 circuits (ASICs), for
 mining, 174
arbitrage, 64
Armory Bitcoin Client, 41
ASICs (application specific integrated
 circuits), for mining, 174
ask order, 63
asymmetric key cryptography, 133
asynchronous programming, 221
ATMs, Bitcoin, 62
Austrian economics, 126
authentication
 password for, 40
 two-factor, 36, 53–54
Authy app, 58–59
average net worth, 121–122

B

Back, Adam, 120
bank account
 linking to Coinbase, 59–60
 linking to exchange, 65
bid orders, 63
BigInteger class (Java), 235
BIP38 encryption, 40
BIPs (Bitcoin Improvement
 Proposals), 40
bitaddress.org, 38
Bitcoin, 1
 in 2030, 199–212
 beginnings, 112–116
 benefits of using, 3
 cap on total supply, 26
 complexity of, 4–5, 27–29
 cryptocurrencies as side
 chains, 121
 energy costs of, 124–125
 future role, 121–123
 how it works, 8–9
 motive for creating, 2–3
 potential of, 116–127
 risk of destruction, 118–119
 safety and security, 31, 61
 units, 9–10
 value growth, 114, 116
Bitcoin addresses, 10–11, 139
 generating
 with Bitcoin wallet
 program, 19
 with master public key, 190
 sharing, 156
 SPV wallets vs. full wallets,
 193–195
Bitcoin ATMs, 62
Bitcoin classes (Java), 231

- bitcoin.conf* file, 218
- Bitcoin Core, 38, 214
 - initializing connection, 220–221
 - JSON-RPC API, 222
 - programming techniques and, 217–218
 - starting, 218–219
 - version 0.1, 113
- bitcoind, 214
 - programming techniques and, 217–218
- Bitcoin exchanges, 52
 - intermediaries, 53–54
 - live exchanges, 71
- Bitcoinians, 53
- Bitcoin Improvement Proposals (BIPs), 40
- BitcoinJ, 226
 - exception types, 239
 - installing, 227–228
 - issues for wallets, 239–240
- Bitcoin network, 169
 - code for connecting, 233–234
- Bitcoin sellers, finding, 67–68
- Bitcoin software applications
 - in JavaScript, 217
 - security notes on programming, 215–216
 - writing approaches, 214–215
- Bitcoin wallets. *See* wallets
- BitPay, 214
- Bitrated, 70
- Bitstamp, 64
- BitTorrent, 119, 127
- black hat hacker, 216
- blind signatures, 111
- block
 - anatomy of, 171–175
 - number of transactions included in, 180
- blockchain, 19–26, 96, 165
 - distribution, 138
 - forking, 23–25
 - importance of, 211
 - initializing, 232
 - lottery, 21–23
 - orphaned, 24–25
 - reasons for, 232–233
 - recording transactions, 161, 170
 - size of, 191
 - storing, 33
- Blockchain.info, 37
- block depth, 25
- block difficulty, 172–173
- block hash, 138
- block header, 171
 - data in, 172
 - and SPV wallets, 192, 193
- BlockStoreException exception type, 239
- Bosselaers, Antoon, 140
- brain wallets, 45–46
- broadcast-only node, 169
- BTC, 9
- BTC China, 64
- BTC-E, 64
- BTCquick, 53
- Buffett, Warren, 110
- buttonwood exchanges, 71
- buying bitcoins, 49–71
 - with Coinbase, 58–61
 - from currency exchange, 62–66
 - methods, 51–52
 - from middleman, 52–57
 - person-to-person, 67–71
- bye-bye-money program, 236–239
 - ensuring money transmission, 238
 - running, 238–239
- Byzantine Generals’ Problem, 2–3, 164–165

C

- C#, 226
- C++, 226
- calculus, 211
- callback function, 221
- cap on total bitcoin supply, 26
- Cavirtex, 64
- change address, 187
- charities, accepting bitcoins, 18
- Chaum, David, 111
- Circle, 53
- client.getBalance function, 222
- client.listTransactions function, 223
- client-server architecture, 119

- Coinbase, 36, 53
 - buying bitcoins with, 58–61
 - linking bank account to, 59–60
 - registering at, 58
- coin control, 196
- cold storage, 47
 - vs. hot storage, 33–34
- collision, hash functions, 132
- colored coins, 205, 206
- comic
 - on altcoins, 181–184
 - on Bitcoin, 73–108
- commodities, spread for, 65–66
- computer viruses, threat to
 - wallets, 216
- confirmed payments, security, 194
- confirming transactions
 - in *Hello Money!* app, 222
 - infinite loop of, 164
- contracts, 55–56
- convenience, of storage, 35
- credit cards, 111, 112
 - vs. Bitcoin transactions, 57
 - issuers, 125
- cross-domain restrictions, 217
- cryptocurrencies, 129
 - competition with Bitcoin, 119–121
- cryptography, 129–159
 - Bitcoin need for, 137–139
 - elliptic curve, 141
 - methods in Bitcoin, 139–141
 - overview, 130–137
 - and rounding errors, 151
 - security for Bitcoin, 157–158
- currencies
 - Bitcoin advantages over existing, 117–118
 - converting to bitcoins, 55–57
 - decentralized, 1
 - ideal, 117
 - stateless, 2
- currency codes, standard for, 9_n
- currency exchanges, 50
 - buying bitcoins from, 62–70
 - opening, 114
 - transferring dollars to account, 65
- Cybercash, 111

D

- Data Universal Numbering Service (DUNS), 214_n
- decentralization, in mining, 179–180
- decoding, cryptography, 134
- decrypting messages, 130
- deflation, dangers of, 126
- <dependencies> section, 229
- deterministic key generation, 187–190
 - combining with watch-only wallet, 189
- difficulty target, 171
- DigiCash, 111
 - bankruptcy, 112
- digital currencies, 1, 64
 - dangers of decentralized, 123–127
 - discussions on government role, 116
 - history, 110–112
- digital signatures, 11, 91, 131–132, 135–136
 - authorizing transactions with, 137–138
 - using elliptic curves, 154–155
- discounts, for limit orders, 66
- discrete logarithm, 131–132
- distributed autonomous corporations, 208
- distributed computing projects, Bitcoin as largest, 115
- distribution of bitcoins, 162
- divisibility, of currency, 117
- DnsDiscovery class, 234
- Dobbertin, Hans, 140
- dollar bill, life span of, 118
- dollars, converting to bitcoins, 55–57
- double SHA256 hash, 171
 - security and, 156
- double spending, 25, 167
- Draper, Adam, 110
- DUNS (Data Universal Numbering Service), 214_n
- durability, of currency, 118

E

- e-cash, 111
- ECDSA (elliptic curve digital signature algorithm), 146–153
 - signing Bitcoin transaction with, 153–156
 - verifying signature with, 155
- e-commerce, building app using, 214
- e-gold, 112
- Electrum wallet, 14–16, 38, 188
- elliptic curve cryptography, 141
 - calculating sum of adding two points, 149
 - pseudocode for summation and multiplication, 158–159
- elliptic curve digital signature algorithm (ECDSA), 146–153
 - signing Bitcoin transaction with, 153–156
 - verifying signature with, 155
- encoding, cryptography, 134
- encryption, 130
 - BIP38, 40
 - paper wallets, 39–40
 - password for, 40
- energy costs, of Bitcoin, 124–125
- error handling, Bitcoin programming, 239
- escrow services, 68, 69
 - face-to-face bitcoin purchase with, 69–71
 - face-to-face bitcoin purchase without, 68
 - setting up, 70
- exchange intermediary, Coinbase as, 58
- exec-maven-plugin plug-in, 229
- ExecutionException exception type, 239

F

- face-to-face bitcoin purchases
 - with escrow, 69–70
 - problems, 69
 - without escrow, 68

- fees, 26–27, 170, 238
 - for Bitcoin transaction, 18
 - for currency exchange, 63
 - for middleman, 53
- field programmable gate arrays (FPGAs), for mining, 174
- Finney, Hal, 113
- first bits scheme, 10*n*
- FPGAs (field programmable gate arrays), for mining, 174
- fragmented private keys, and multi-signature addresses, 41–42
- fraud prevention, 125
- Freenet, 127
- Friedman, Milton, 110
- full node, 191
- full payment verification, 191
- full wallets, 187
 - vs. SPV wallets, 193–195
- fungibility, of currency, 118

G

- generator point, elliptic curve cryptography, 152
- genesis block, 113, 165
- German mark, 2*n*
- Git, installing, 227
- git checkout command, 228
- Gnutella, 119, 127
- gold, wealth stored as, 121
- gold coins, 1
- goods, first exchange for bitcoins, 114
- Go programming language, 226
- government
 - digital currency companies and, 111–112
 - risk of Bitcoin destruction by, 119
 - stability, and Bitcoin, 126–127
- graphics-processing units (GPUs), for mining, 174

H

- hacker theft, likelihood of, 38
- hardware, for mining, 174–175
 - 2030 requirements, 202
 - energy efficiency of, 178
 - profitability threshold curves for comparing, 179

- hardware wallets, 42–43
- hash, 98, 132–133
 - of transactions in block, 172
- hash functions, 131
 - for verifying information, 132–133
- hash rate
 - projecting future, 177
 - theoretical limits, 178–179
- Hayek, Friedrich, 126
- health of network, SPV wallets vs. full wallets, 195
- heavyweight wallets, 191
- hellomoney.js* file, 220
- Hello Money!* program, 217–218, 220–222
- hello-money starter project
 - creating, 228–229
 - declarations, 231
 - hook for detecting money arrival, 234
 - running and testing, 235–236
 - writing code, 230–235
- hierarchical deterministic wallets, 190
- Hill, Austin, 120
- history of Bitcoin, 112–116
- homebrew (command-line tool), 219
- hosted wallets
 - online services, 36
 - vs. personal wallets, 34–35
- hot storage, 47
 - vs. cold storage, 33–34
- hot wallets, personal, 37–38
- human-readable Bitcoin
 - addresses, 10*n*
- hybrid wallets, 187

I

- illegal activity, Bitcoin and, 124
- impedance mismatch, 57
- importing private key, 17, 39, 193, 194–195, 237
- installing SPV wallets vs. full wallets, 193
- integer factorization, 131
- Internet bubble, 120
- InterruptedException exception
 - type, 239

- irreversibility, of transactions,
 - 25–26, 56
 - superiority of, 57

J

- Java, 226
 - initializing objects, 231–233
 - installing, 226–227
- java.io.File class, 231
- Java JDK (Java Development Kit), 226
- java.matho.BigInteger class, 231
- JavaScript, 213–223
 - preparing machine for, 218–219
 - writing Bitcoin program in, 217–218
- jelly-filled donut incident, 141–156
- JSON-RPC API (JavaScript Object Notation - Remote Protocol Call), 222
 - limitations of writing Bitcoin programs using, 223
- JSON-RPC protocol, 214

K

- Kaminsky, Dan, 118
- Keynesian economics, 126
- Kienzle, Jörg, 110–111
- Koblitz curve, 151
- Kraken, 64
- Krugman, Paul, 117

L

- Landauer limit, 157
- laptops, private keys on, 44
- ledger, 11
- length extension, 171*n*
- liability, for stolen bitcoins, 34
- lightweight wallets, 192
- limit orders, 66
- Linux
 - installing Git, 227
 - installing Maven, 227
 - OpenJDK version of Java, 227
 - setting up Bitcoin Core server, 219
- live Bitcoin exchanges, 71
- LocalBitcoins.com, 67, 68
 - escrow service, 70

M

Mac OS

- installing Git, 227
- installing Maven, 227
- setting up Bitcoin Core server, 219

man-in-the-middle attacks, 216

market orders, 65–66

MasterCard, 112

master private key, 188

master public key, 188

- generating Bitcoin address with, 190

Maven

- empty starter project
 - created with, 228
- installing, 227

mBTC (millibitcoins), 9

MD5 (message digest algorithm), 132

meeting places, for Bitcoin

- transactions, 68

MemoryBlockStore function (bitcoinJ), 237

merchant services, 214

Merkle trees, 192

mesh networks, 169

message digest algorithm (MD5), 132

microbitcoins (μ BTC), 9

middleman, buying bitcoins from, 52–57

Miller-Rabin primality test, 90

millibitcoins (mBTC), 9

mining, 5, 20, 26–27, 96, 99, 161–180

- in 2030, 201–202

- decentralization of, 179–180

- difficulty of, 173

- distributing new currency with, 167–168

- hardware, 174–175

 - 2030 requirements, 202

 - energy efficiency of, 178

 - profitability threshold curves for comparing, 179

- need for, 162–168

- nodes, 170

- pooled, 175–176

- practicality, 50

- preventing attacks with, 166–167

- process for, 168–176

 - for profit, 176–177

 - proof-of-work in, 138–139

 - solving a block, 171

modular arithmetic, 131*n*

“m of n” private key, 42

money laundering, 112–113

Moore’s law, 179*n*

Moxie Jean, 67

Multibit, 38

multi-signature addresses, and

- fragmented private keys, 41–42

multi-signature transactions, 57, 69–70

mvn install command, 230

My Wallet Service, 37

N

Nakamoto, Satoshi, 3, 110, 211

- identity, 113

- last comment, 114

- white paper on Bitcoin, 112

network effect, 120

NetworkParameters structure, 232

newbiecoins.com, 13

newly minted bitcoins, 26–27

Newton, Isaac, *Principia*, 210–211

node-bitcoin, installing, 218

Node.js library, 217, 221

- installing, 218

Node Package Manager, 218

nodes

- broadcast only, 169

- full, 191

- relay, 170

nominal deflation, 126

nonprofit organizations, accepting

- bitcoins, 18

NXT, 125

O

off-chain transactions, 201

offline transaction signing, 40–41

onCoinsReceived function, 234–235

online wallet services

- hosted, 36

- personal, 34, 37

- Oracle Corporation, 226
- orders, placing to buy bitcoins, 65
- order of curve, elliptic curve
 - cryptography, 152–153
- orphaned blocks, 24–25

P

- paper money, color copiers as
 - threat, 110
- paper wallets, 39
 - encrypted, 39–40
- passwords, 14, 40
 - for brain wallet, 45
 - function of, 40
 - loss of, 37
- Peercoin, 125
- PeerGroup object, 233–234, 240
- peer-to-peer architecture, 119
- pegging, 120
- pending transaction, 18
- Perrig, Adrian, 110–111
- personal wallets
 - vs. hosted wallet, 34–35
 - hot storage, 37–38
 - online services, 37
- person-to-person bitcoin purchases, 52, 67–71
- point multiplication, 150, 158–159
- point-of-sale terminals, watch-only
 - wallet for, 187
- polling, Bitcoin programming, 223
 - pom.xml* file, 229, 236–237
- pooled mining, 175–176
- portability, of currency, 117
- Preneel, Bart, 140
- price discovery process, 120
- privacy, 11*n*
 - and criminals, 124
 - multiple addresses and, 12
- private currencies, 2
- private key, 11–12, 150
 - compromise of, 41
 - extra protection for, 139
 - fragmented, and multi-signature
 - addresses, 41–42
 - generating, 37
 - importing, 237
 - master, 188

- memorizing, 45
- parable on, 141–145
- reversing function of, 136
- security for, 39, 186
- signing transaction with, 156
- SPV wallets vs. full wallets, 194
- storing, 33

- profit, mining for, 176–177
- programming languages, for Bitcoin
 - network connection, 225–226
- proof-of-stake, 125
- proof-of-work, 125, 166
 - and blockchain, 165
 - in mining, 138–139
- protecting bitcoins, 61. *See also* security
- protocol, for Bitcoin, 112
- public information, transactions as, 11
- public key, 150
 - encryption, 91
 - master, 188
 - parable of, 141–145
 - reversing function of, 136
 - sharing, 156
- public key cryptography, 133–135
- public/private key pair, creating with
 - ECDSA, 154
- pushing, Bitcoin programming, 223
- Python, 226

Q

- quick response (QR) codes, for
 - Bitcoin address, 10

R

- Race Integrity Primitives Evaluation
 - Message Digest (RIPEMD), 139–141, 188
- radical decentralization, 126
- random key generation, 187–190
- randomness, for generating Bitcoin
 - address, 39
- relay node, 170
- RelayRides, 67
- remote servers, Electrum
 - connection to, 15

- retailers, acceptance of Bitcoin, 116
- reversible transactions, 55–56
- rewards, 170
 - from Bitcoin-mining lottery, 22
 - for transaction processing, 26
- RIPMD (Race Integrity Primitives Evaluation Message Digest), 139–141, 188
- risks, to Bitcoin, 117–121
- Rivest, Ron, 133–134
- rounding errors, 235
 - and cryptography, 151
- RSA encryption, 133–134, 137
- Ruby, 226

S

- safety, of storage, 35
- satoshi (bitcoin unit), 9
- SatoshiLabs, 43
- Satoshi Square, 71
- savings, Bitcoin for, 121–122
- scarcity, of currency, 118
- Sean's Outpost, 18
- Secure Hash Algorithm (SHA), 139–141, 188
 - ASIC optimization to calculate, 174
- security, 14, 118–119
 - of Bitcoin exchanges, 63
 - confidence in, 216–217
 - double hash scheme and, 156
 - SPV wallets vs. full wallets, 193–194
 - of storage, 35
- seed, in Electrum, 14, 15
- sending money
 - from Bitcoin address, 236–239
 - code for, 238
- SendRequest object, 238
- settlement period, 55, 56
- SHA (Secure Hash Algorithm), 139–141, 188
 - ASIC optimization to calculate, 174
- Shamir, Adi, 133–134
- Shamir's Secret Sharing method, 42
- shares, of mining reward, 176
- side chains, 121
- Silk Road website, 124

- simplified payment verification (SPV), 191, 233
 - vs. full wallets, 193–195
- single key generation wallet programs, 188
- smartphones
 - private keys on, 44
 - wallets on, 192
- software as a service, 34
- speed of payments, SPV wallets vs. full wallets, 193
- spending bitcoins, 17–19
- SPV (simplified payment verification), 191, 233
 - vs. full wallets, 193–195
- SPVBlockStore object, 232, 233
- stateless currencies, 2
- storage, Bitcoin, 31–47
 - choosing method, 46–47
 - hot vs. cold, 33–34
 - of large amounts of bitcoins, 38–42
 - private key, 33
 - safety, security, and convenience, 35
 - of small amounts of bitcoins, 35–38
 - SPV wallets vs. full wallets, 194
 - Trezor, 43–45
- summation, pseudocode for, elliptic curve cryptography, 158–159
- symmetric key cryptography, 133
- synchronization, SPV wallets vs. full wallets, 193

T

- Takhteyev, Yuri, 112*n*
- tangent to curve, elliptic curve cryptography, 150
- thick wallets, 191
- thin wallets, 192
- third-party service provider, as bank, 33
- timestamp, for block, 172
- Tor, 127
- trade volume, of exchange, 63
- transaction confirmation, 25
- transaction fees. *See* fees

- transaction history, verifying
 - validity, 138
- transactions
 - authorizing with digital
 - signatures, 137–138
 - full vs. simplified payment
 - verification, 191–195
 - information in, 138
 - off-chain, 201
 - ordering, 166
 - potential, in 2030, 201
 - signing
 - with ECDSA, 153–156
 - offline, 40–41
 - offline vs. online, 186–187
 - with private key, 156
- transferring dollars to exchange
 - account, 66
- Trezor, 43–45
- true blockchain, 24
- true ledger, 166
- trust, 110, 111
- two-factor authentication, 36, 53–54
 - setting up in Coinbase, 58–59

U

- unit of account, 123
- units, Bitcoin, 9
- unspent output, 196

V

- valid transaction, 191
- vendor APIs, 214–215
- Visa, 112
- volatility, of Bitcoin, 120

W

- wallet file, 13, 33, 186
- `Wallet.loadFromFile()` function, 240
- walletNotify* feature, 223

- wallets, 12–19, 28–29, 185–198
 - acquiring bitcoins, 16
 - BitcoinJ issues, 239–240
 - brain, 45–46
 - creating empty, 232
 - future changes, 197
 - generating, 38–42
 - getting bitcoins into, 17
 - hardware, 42–43
 - online hosted services, 36
 - paper, 39
 - personal vs. hosted, 34–35
 - running on autopilot, 214
 - selecting, 197
 - software design fundamentals,
 - 186–195
 - features, 195–196
 - offline vs. online transaction signing, 186–187
 - random vs. deterministic key generation, 187–190
 - transferring coins from Coinbase
 - wallet to, 61
 - virus threat to, 216
 - watch-only. *See* watch-only wallet
- `Wallet.saveToFile()` function, 240
- watch-only wallet, 186
 - combining deterministic key generation with, 189
 - math supporting, 189–190
 - full vs. SPV, 191–195
 - for point-of-sale terminals, 187
- Windows development environment,
 - JavaScript on, 218–219

X

- XBT, 9

Z

- zero point, elliptic curve
 - cryptography, 152–153