

Index

A

Abstraction layers, 17
admin.addPeer() command, 331
Advanced Encryption Standard (AES)
 AddRoundKey, 52
 AES-128, 48
 8-bit byte, 49
 block size, 48
 encryption and decryption
 process, 49–50
 key expansion, 53–54
 MixColumns, 51
 NIST, 54
 processing steps, 48
 round function, 50
 ShiftRows, 51
 state array, 49
 state words, 49
 SubBytes, 50
 substitution-permutation
 network, 48–49
Amazon, 14, 23, 184
Application Binary Interface (ABI),
 262–263, 297, 302, 308, 310,
 336, 337
Application layer, 19, 25

Asymmetric key cryptography
 App stores, 80
 authentication, 79
 code examples, 95–97
 confidentiality, 79
 digital signatures, 78
 DSA, 86–88
 ECC (*see* Elliptic curve
 cryptography (ECC))
 ECDSA, 93–95
 private key, 80
 public key, 79–81
 RSA algorithm
 encryption/decryption, 84–86
 generation of key pairs, 82–84
 modular arithmetic, 82
 vs. symmetric key
 cryptography, 102–104
 text message, Alice to Bob, 78
Autonomous Decentralized Peer-To-Peer
Telemetry (ADEPT), 27

B

Banking era, 152
Bitcoin, 3, 149
 bitcoinjs, 215, 272
 block explorer API, 272

INDEX

Bitcoin (*cont.*)

- block structure
 - difficulty target, 165–168
 - header components, 162–163
 - field and size, 161
 - Merkle trees, 163–165
- data structure, 159
- dawn, 153–154
- defined, 154–157
- Ethereum, 220–221
- full nodes, 209–210
- genesis block
 - chainparams.cpp, 169
 - hash information, 171
 - transaction information, 170
- mining, 22
- orphan blocks, 160
- PoW, 22
- smart contracts, 20
- SPVs, 210, 212
- transaction, Bitcoin test
 - network
 - addOutput method, 279
 - broadcast transaction, 281–282
 - transaction.addInput method, 279
 - get test net Bitcoins, 275
 - hex string, 280
 - keypairs creation, 274
 - sender's unspent outputs, 276–278
 - setup and initialization, bitcoinjs library, 273–274

- sign transaction inputs, 280
- wallets, 212–215
- working with, 157–158

Bitcoin network

- block propagation, 193–194
- consensus and block mining (*see* Block mining)
- discovery, new node, 174–178
- full/lightweight nodes, 173
- on Internet, 172
- SPV, 173–174
- transactions, 179–184

Bitcoin scripts

- CheckSig, 207
- defined, 204
- formation of combined validation, 205
- granular components, 200
- input and output code, 203–204
- practical example, 202
- ScriptPubKey, 201, 203, 207
- ScriptSig, 201
- stack-based implementation, 206–207
- transaction fields, 199
- transactions revisited, 196–198

BitcoinJ, 215

Blockchain

- advantages, 1
- applications
 - actors, handle requests, 137
 - backend database, 135
 - Bitcoin node, 135–136
 - centralized web server, 135

- cloud-empowered
 - blockchain system, 136–137
- cloud services, 136, 138
- consensus algorithms, 137
- DApps, Ethereum network, 138
- development, 269–270
- hybrid, 138
- interaction, 271
 - public blockchain, 136
- banking system, 2–3
- Bitcoin (*see* Bitcoin)
- business problems and situations, 34
- Byzantine Generals' Problem, 33
- centralized system, 33
- components, 32
- computer science engineering (*see* Computer science engineering)
- core, 32
- cryptocurrency
 - implementations, 32
- cryptography (*see* Cryptography)
- data structure, 9, 123
- decentralized and peer-to-peer solution, 6–7
- description, 31
- distributed consensus
 - mechanisms, 130–131
- fundamentals, 122
- game theory (*see* Game theory)
- handcrafting transactions, 312–313
- intermediary *vs.* peer-to-peer transaction, 4–5
- offerings, 24
- PBFT, 134–135
- PoS, 133–134
- PoW, 131–133
- properties
 - auditability, 127
 - consistent state of ledger, 127
 - democratic, 125
 - double-spend resistant, 126
 - forgery resistant, 125
 - immutability, 125
 - resilient, 127
- real-world business problems, 31
- scalability
 - Bitcoin adoption, 139
 - centralized system, 139
 - consensus protocols, 139
 - database sharding, 143–145
 - disruptive technologies, 139
 - off-chain computation, 140–143
 - public and private Blockchains, 140
 - transactions, 139
- scenarios, 123
- transactions, 127–129, 312
- use cases, 26–27

INDEX

- Block ciphers, 40–41
 - Block mining
 - ballpark values, 190
 - block header, 189
 - block reward, 187
 - coin creation, 187
 - cryptographic security, 185
 - defined, 184
 - halving process, 187
 - hash and target value, 190
 - incentivization mechanism, 192
 - miners, 188
 - nodes, 188
 - orphaned blocks, 193
 - PoW, 185, 191
 - transaction fees, 186
 - valid block, 190
 - Bureaucratic system, 3
 - Business transaction, 4
 - Byzantine Generals' Problem, 110–112, 114
- ## C
- Centralized systems
 - advantages, 14
 - vs.* decentralized systems, 11–14
 - limitations, 14, 23–24
 - Coinbase transaction, 179
 - Computer science engineering
 - blockchain
 - block-1234, 116
 - block structure, 117
 - data structure, 114
 - genesis block, 115
 - hash pointer, 114–115
 - parent block, 115
 - SHA-256, 116
 - Merkle trees, 117–122
 - Consensus layer, 22
 - Contract.deploy method, 303
 - Cryptography
 - advanced mathematical techniques, 34
 - asymmetric key (*see* Asymmetric key cryptography)
 - authentication, 35
 - ciphertext, 35–36
 - confidentiality, 35
 - data integrity, 35
 - Diffie-Hellman key exchange, 98–101
 - encryption techniques, 35
 - hash functions (*see* Hash functions)
 - non-repudiation, 35
 - plaintext, 35
 - steps, 36
 - symmetric key (*see* Symmetric key cryptography)
 - transactions, 33
- ## D
- Database sharding, 143–145
 - Data Encryption Standard (DES)
 - 64-bit block size, 43

- cryptography, 44
 - Feistel cipher, 43, 45–47
 - key generator, 44
 - limitations, 48
 - Moore’s law, 43
 - round function, 47
 - Decentralized applications (DApps)
 - architecture
 - public nodes *vs.* self-hosted nodes, 315–316
 - servers, 316
 - blockchain-based, 268
 - client application, web3
 - getPoll function, 371
 - html file and scripts, 360
 - JavaScript functions, 371
 - polling web application view, 366
 - send on vote function, 372
 - smart contract interaction code, 366
 - transaction, smart contract function, 372–373
 - voted event, 373–374
 - web3.eth.Contract submodule, 371
 - private Ethereum network (*see* Private Ethereum network)
 - smart contract (*see* Smart contract, DApp)
 - voting system, 269
 - Decentralized applications (DApps), 138
 - Decentralized systems
 - advantages, 15, 24
 - vs.* centralized systems, 11–14
 - limitations, 15
 - peer-to-peer system, 16
 - Diffie-Hellman key
 - exchange, 98–101
 - Digital signature algorithm (DSA), 62, 86–88
- E**
- Elliptic curve cryptography (ECC)
 - 160-bit ECC key, 88
 - characteristics, 89–92
 - discrete logarithm problem, 88
 - domain parameters, 92
 - mathematical equation, 88
 - shapes, 88
 - Elliptic Curve Diffie-Hellman (ECDH), 93
 - Elliptic Curve Digital Signature Algorithm (ECDSA)
 - key generation, 93
 - vs.* RSA, 93
 - sender and receiver, 93
 - signature generation, 94
 - signature verification, 94–95
 - Ethereum blockchain, 219
 - accounts
 - advantages, 232–233
 - Contract Accounts, 228
 - EOAs (*see* Externally Owned Accounts (EOAs))

INDEX

Ethereum blockchain (*cont.*)

- state, 233–235
- UTXOs advantages, 231–232

Bitcoin to, 220–221

block metadata, 226

consensus-subsystem
information, 227

data references, 226

data structure, 225

decentralized applications,
221, 222

design philosophy, 223–224

ecosystem

- DApp, 264

- development components,
265

- limitations, 263

- Swarm, 264

- Whisper, 264

EVM, 222, 257–262

gas and transaction cost,
248–253

Infura API service, 284

Merkle Patricia tree, 237–239

mining, 22

PoW, 22

RLP encoding, 239

ropsten test network

- library and connection,
284–285

- preparation, 287–288

- send transaction, 290–292

- set up Ethereum accounts,
285–286

- sign transaction, 288–289

- testnet faucets, sender's
account, 286–287

smart contracts, 20, 253–254

- application, 256

- blocks, 255

- compilation, 297

- contract creation, 256–257

- deploy, 302

- executing, 309–311

- Remix IDE, 294–295

- solidity programming
language, 293

- transaction, 295, 297

- voting application, 255

software development and
deployment, 223

state transaction function,
245–247

transaction and message
structure, 240–244

transaction execution
information, 227

trie usage, 236

Ethereum virtual machine
(EVM)

- ABI, 262–263

- absolute determinism, 258

- easy security, 258

- JVM, 222

- memory, 261–262

- native operations, 258

- P2P network, 259
- simplicity, 257–258
- smart contract deployment and usage, 259–260
- space optimization, 258
- stack, 262
- storage, 260–261
- Ethminer, 253
- Execution layer, 20
- Externally Owned Accounts (EOAs), 228
 - to Contract Account transaction, 230–231
 - to EOA transaction, 229

F

- Feistel cipher, 43, 45–47
- Fiat currency, 152
- Financial services market, 25

G

- Game theory
 - Bitcoins, 104
 - blockchain job, 104
 - Byzantine Generals' Problem, 110–112, 114
 - cricket tournament, 104
 - Nash Equilibrium, 107–108
 - prisoner's dilemma, 108–110, 113
 - real-life situations, 104, 106, 113
 - sport event, 106

- strategies, 105
- vegetables, 105
- zero-sum games, 112–113
- Government sectors, 28

H

- Handcrafting transactions, 312–313
- Hash functions
 - applications, 73
 - basic form, 56
 - Bitcoin, 60
 - code examples, 74–75
 - core properties, 56
 - hash value, 56
 - information security
 - applications, 55
 - message digest (MD)
 - family, 62
 - puzzle friendliness, 60
 - RIPEDM, 67
 - search puzzle, 61
 - security properties
 - collision resistance, 57–58
 - pre-image resistance, 58–60
 - SHA (*see* Secure Hash Algorithm (SHA))
- Hyperledger, 20, 24, 117, 134, 139

I, J

- Initial Coin Offering (ICO), 26
- Internet Engineering Task Force (IETF), 74

INDEX

K

Keypairs, 274

L

Layers

- abstraction, 17
- application, 19
- consensus, 22
- execution, 20
- propagation, 21–22
- semantic, 20–21

M

Merkle trees, 21, 117–122

Message authentication code
(MAC), 55, 76–77

Mining, 156

Mist wallet, 265

Monetary transactions, 9–11

Money

- banking era, 151–152
- fiat currency, 152, 153
- gold and silver metals, 151
- Internet, 153
- pimitive barter system, 150

N, O

Nash Equilibrium, 107–108

National Institute of Standards and
Technology (NIST), 54

National Security Agency (NSA), 62

P, Q

PBFT, *see* Practical Byzantine Fault
Tolerance (PBFT)

Pimitive barter system, 150

PoS, *see* Proof of Stake (PoS)

PoW, *see* Proof of Work (PoW)

Practical Byzantine Fault

Tolerance (PBFT), 134–135

Prisoner's dilemma, 108–110, 113

Private Ethereum network

account creation, 323

first node

configuration, 327

custom genesis

configuration, 326

geth command, 326–328

genesis.json configuration

file, 324–325

geth data directory, 322

install geth, 321

second node

command, 330

genesis.json

configuration, 329

geth console, peers, 331–332

geth initialize

configuration, 329

geth logs, 333

Proof of Stake (PoS), 133–134

Proof of Work (PoW), 22, 131–133

Propagation layer, 21–22

Pseudorandom number generator
(PRNG), 40, 43, 73

Public key infrastructure
(PKI), 80–81

Public *vs.* private
blockchains, 313–314

R

RACE Integrity Primitives
Evaluation Message Digest
(RIPEMD), 67

Regular transactions, 179

S

ScriptPubKey, 198

Secure Hash Algorithm (SHA)

DSA, 62

NSA, 62

SHA-1, 62, 64

SHA-2, 63–64

SHA-3

cryptographic hash
functions, 68

Merkle-Damgård
construction, 68

NIST, 67

sponge construction, 68–70,
72

state array representationin,
71

variants, padding, 69

SHA-256 and SHA-512, 65–66

versions, 62

Semantic layer, 20–21

sendSignedTransaction
function, 290

Simplified Payment Verification
(SPV), 173

signTransaction function, 289–290

Smart contract, Ethereum DApp

client applications

html code, 360

interaction code, 366

polling web application

view, 366

transaction, 372

creation

ABI, 337, 339

byte code, 339

polling functionality, 334

preceding code snippet, 335

Remix online Solidity editor,
337

solidity code snippet, 334

voting functionality, 336

deploying

private network, 345

web3 library and
connection, 345

Smart software engineering, 34

Solidity programming language,
293

Stock transaction, 5–6

Stream ciphers, 39–40

Supply chains, 28

Symmetric key cryptography

AES (*see* Advanced Encryption
Standard (AES))

INDEX

Symmetric key cryptography (*cont.*)

- vs.* asymmetric key
 - cryptology, 102–104
- block ciphers, 40–41
- ciphertext, 37
- DES (*see* Data Encryption Standard (DES))
- file transfer protocols, 38
- Kerckhoff’s principle and XOR
 - function, 38–39
- limitations, 55
- MAC and HMAC, 76–77
- one-time pad, 42–43
- sender and receiver, 37
- “shared secret”, 37
- stream ciphers, 39–40

T, U, V

- transaction.sign function, 280
- Transmission Control Protocol/
 - Internet Protocol (TCP/IP),
2, 17
- Truffle, 265

W, X, Y

- Web3.js, 265
- The Wisdom of Crowds, 27
- World Wide Web (WWW), 2

Z

- Zero-sum games, 112–113