

Bitcoin

Grundlagen und Programmierung

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Vorwort	15
1 Einführung	27
Geschichte des Bitcoins	30
Erste Schritte	31
Wahl einer Bitcoin-Wallet	31
Schnelleinstieg	33
Wiederherstellungscodes (Recovery Codes)	34
Bitcoin-Adressen	35
Bitcoin empfangen	36
Ihr erster Bitcoin	36
Den aktuellen Bitcoin-Preis ermitteln	38
Bitcoin senden und empfangen	38
2 Wie Bitcoin funktioniert	41
Bitcoin-Übersicht	41
Kaufen im Onlineshop	42
Bitcoin-Transaktionen	43
Inputs und Outputs von Transaktionen	44
Transaktionsketten	44
Wechselgeld	45
Coin-Auswahl	46
Gängige Transaktionsformen	46
Eine Transaktion konstruieren	48
Die richtigen Inputs	48
Die Outputs erzeugen	48
Die Transaktion zur Blockchain hinzufügen	49
Bitcoin Mining	50
Die Transaktion einlösen	53

3	Bitcoin Core: die Referenzimplementierung	55
	Von Bitcoin zu Bitcoin Core	55
	Bitcoin-Entwicklungsumgebung	57
	Bitcoin Core aus dem Quellcode kompilieren	57
	Wahl einer Bitcoin-Core-Release	58
	Den Bitcoin-Core-Build konfigurieren	59
	Die Bitcoin-Core-Executables erzeugen	61
	Eine Bitcoin-Core-Node betreiben	62
	Den Bitcoin-Core-Node konfigurieren	63
	Bitcoin-Core-API	67
	Informationen zum Status von Bitcoin Core erhalten	68
	Transaktionen untersuchen und decodieren	69
	Blöcke untersuchen	71
	Die Bitcoin Core API nutzen	72
	Alternative Clients, Bibliotheken und Toolkits	75
	C/C++	76
	JavaScript	76
	Java	76
	Python	76
	Go	76
	Rust	76
	Scala	76
	C#	77
4	Schlüssel und Adressen	79
	Public-Key-Kryptografie	80
	Private Schlüssel	81
	Kryptografie mit elliptischen Kurven	82
	Öffentliche Schlüssel	85
	Output- und Input-Skripte	87
	IP-Adressen: die ursprünglichen Bitcoin-Adressen (P2PK)	87
	Altadressen für P2PKH	89
	Base58Check-Codierung	91
	Komprimierte öffentliche Schlüssel	94
	Alte Pay-to-Script-Hashes (P2SH)	96
	Bech32-Adressen	98
	Probleme mit Bech32-Adressen	101
	Bech32m	101
	Formate privater Schlüssel	105
	Komprimierte private Schlüssel	106
	Fortgeschrittene Schlüssel und Adressen	108
	Vanity-Adressen	108
	Paper-Wallets	110

5	Wallet-Recovery	113
	Unabhängige Schlüsselgenerierung	113
	Deterministische Schlüsselgenerierung	114
	Ableitung öffentlicher Child-Schlüssel	116
	Hierarchisch-deterministische (HD-)Schlüsselgenerierung (BIP32) ...	117
	Seeds und Recovery-Codes	118
	Nichtschlüsseldaten sichern	122
	Schlüsselableitungspfade sichern	123
	Details des Wallet-Technologiestacks	125
	BIP39-Recovery-Codes	126
	Eine HD-Wallet aus dem Seed-Wert erzeugen	132
	Einen erweiterten öffentlichen Schlüssel in einem Webshop nutzen	137
6	Transaktionen	143
	Eine serialisierte Bitcoin-Transaktion	143
	Version	145
	Erweiterter Marker und Flag	146
	Inputs	146
	Länge der Input-Liste der Transaktion	147
	Outpoint	148
	Input-Skript	150
	Sequenz	150
	Outputs	154
	Outputs-Zähler	154
	Menge	154
	Output-Skripte	156
	Witness-Struktur	157
	Zirkulare Abhängigkeiten	158
	Transaktionsverformbarkeit durch eine dritte Partei	158
	Transaktionsverformbarkeit durch eine zweite Partei	159
	Segregated Witness	160
	Serialisierung der Witness-Struktur	162
	Locktime	162
	Coinbase-Transaktionen	163
	Gewicht und Vbytes	164
	Veraltete Serialisierung	166
7	Autorisierung und Authentifizierung	167
	Transaktionsskripte und Skriptsprache	167
	Turing-Unvollständigkeit	168
	Zustandlose Verifikation	168
	Konstruktion von Skripten	168
	Pay-to-Public-Key-Hash	172

Geskriptete Multisignaturen	174
Eine Eigentümlichkeit in der CHECKMULTISIG-Ausführung	175
Pay-to-Script-Hash	176
P2SH-Adressen	178
Vorteile von P2SH	179
Redeem-Skript und Validierung	179
Data Recording Output (OP_RETURN)	179
Einschränkungen von Transaktions-Locktimes	181
Check Lock Time Verify (OP_CLTV)	181
Relative Timelocks	183
Relative Timelocks mit OP_CSV	184
Skripte mit Ablaufsteuerung (Bedingungsklauseln)	184
Bedingungsklausel mit VERIFY-Opcodes	185
Die Ablaufsteuerung in Skripten nutzen	186
Komplexes Skriptbeispiel	188
Segregated-Witness-Output- und Transaktionsbeispiele	189
Upgrade auf Segregated Witness	192
Merkalized Alternative Script Trees (MAST)	194
Pay-to-Contract (P2C)	198
Skriptlose Multisignaturen und Threshold-Signaturen	199
Taproot	201
Tapscript	203
8 Digitale Signaturen	205
Wie digitale Signaturen funktionieren	205
Eine digitale Signatur erzeugen	206
Die Signatur verifizieren	206
Arten von Signatur-Hashes (SIGHASH)	207
Schnorr-Signaturen	209
Serialisierung von Schnorr-Signaturen	215
Schnorr-basierte skriptlose Multisignaturen	215
Schnorr-basierte skriptlose Threshold-Signaturen	217
ECDSA-Signaturen	219
ECDSA-Algorithmus	220
Serialisierung von ECDSA-Signaturen (DER)	221
Die Bedeutung der Zufälligkeit für Signaturen	222
Segregated Witness' neuer Signieralgorithmus	223
9 Transaktionsgebühren	225
Wer zahlt die Transaktionsgebühr?	226
Gebühren und Gebührensätze	227
Angemessene Gebührenraten bestimmen	228
Replace-By-Fee (RBF)	229
Child-Pays-for-Parent (CPFP)	232

Paketweiterleitung (Package Relay)	233
Transaktions-Pinning	234
CPFP-Carve-out und Anker-Outputs	235
Gebühren in Transaktionen einfügen	236
Timelock-Schutz gegen Fee-Sniping	237
10 Das Bitcoin-Netzwerk	239
Arten und Rollen von Nodes	240
Das Netzwerk	240
Compact Block Relay	240
Private Block-Relay-Netzwerke	243
Netzwerkerkundung	244
Full Nodes	248
»Inventar« austauschen	249
Leichtgewichtige Clients	250
Bloomfilter	252
Wie Bloomfilter funktionieren	253
Wie leichtgewichtige Clients Bloomfilter nutzen	256
Kompakte Blockfilter	257
Golomb-Rice Coded Sets (GCS)	258
Welche Daten in einen Blockfilter gehören	260
Blockfilter von mehreren Peers herunterladen	261
Bandbreite reduzieren durch verlustbehaftete Codierung	262
Kompakte Blockfilter nutzen	262
Leichtgewichtige Clients und Privatsphäre	263
Verschlüsselte und authentifizierte Verbindungen	263
Mempools und Waisenpools	264
11 Die Blockchain	267
Struktur eines Blocks	268
Block-Header	269
Blockkennungen: Block-Header-Hash und Blockhöhe	269
Der Genesis-Block	270
Blöcke in der Blockchain verlinken	271
Merkle Trees (Hashbäume)	273
Merkle Trees und leichtgewichtige Clients	277
Bitcoins Test-Blockchains	278
Testnet: Bitcoins Testspielwiese	278
Signet: das Proof-of-Authority-Testnet	280
Regtest: die lokale Blockchain	281
Test-Blockchains zur Entwicklung nutzen	283

12 Mining und Konsens	285
Bitcoin-Ökonomie und Währungsgenerierung	287
Dezentralisierter Konsens	289
Unabhängige Verifikation von Transaktionen	290
Mining-Nodes	291
Die Coinbase-Transaktion	292
Coinbase-Belohnungen und Gebühren	292
Struktur einer Coinbase-Transaktion	293
Coinbase-Daten	294
Den Block-Header aufbauen	295
Mining des Blocks	296
Proof-of-Work-Algorithmus	297
Target-Darstellung	299
Retargeting zur Anpassung der Difficulty	299
Median Time Past (MTP)	301
Den Block erfolgreich schürfen	302
Einen neuen Block validieren	303
Ketten von Blöcken zusammensetzen und auswählen	304
Mining und der Hashing-Wettlauf	305
Die Lösung mit der Extra-Nonce	306
Mining-Pools	306
Konsensangriffe (Hashrate Attacks)	310
Die Konsensregeln ändern	313
Hard-Forks	313
Soft-Forks	317
Entwicklung von Konsenssoftware	323
13 Bitcoins und Sicherheit	325
Sicherheitsgrundsätze	325
Bitcoin-Systeme sicher entwickeln	326
Die Wurzel des Vertrauens	327
Best Practices für den Nutzer	328
Physische Speicherung von Bitcoins	329
Hardware-Wallets	329
Zugriff sicherstellen	329
Risikodiversifizierung	330
Multisignaturen und Kontrolle	330
Überlebensfähigkeit	330

14 Blockchain-Anwendungen	331
Grundbausteine (Primitive)	331
Anwendungen aus Grundbausteinen	333
Colored Coins	334
Single-Use Seals	334
Pay-to-Contract (P2C)	335
Clientseitige Validierung	336
RGB	336
Taproot-Assets	337
Zahlungs- und Zustandskanäle	338
Zustandskanäle – grundlegende Konzepte und Terminologie	339
Einfaches Zahlungskanalbeispiel	341
Vertrauensfreie Kanäle aufbauen	343
Asymmetrisch widerrufliche Commitments	346
Hash Time Lock Contracts (HTLC)	351
Geroutete Zahlungskanäle (Lightning Network)	352
Einfaches Lightning-Network-Beispiel	352
Lightning Network – Transport und Routing	355
Vorteile des Lightning Network	357
Anhang A: Das Bitcoin-Whitepaper von Satoshi Nakamoto	359
Anhang B: Errata zum Bitcoin-Whitepaper	371
Anhang C: Bitcoin Improvement Proposals	377
Index	383