

# Kryptografie in der Praxis

Eine Einführung in die bewährten Tools,  
Frameworks und Protokolle

# DAS INHALTS- VERZEICHNIS

» Hier geht's  
direkt  
zum Buch

# Inhalt

<b>Vorwort</b>	<b>xvii</b>
Danksagungen	xxi
Über dieses Buch	xxii
<b>Über den Autor</b>	<b>xxvii</b>
<b>Teil A Primitive: Die Elemente der Kryptografie</b>	<b>1</b>
<b>1 Einführung</b>	<b>3</b>
1.1 Kryptografie sichert Protokolle	4
1.2 Symmetrische Kryptografie: Was ist symmetrische Verschlüsselung?	5
1.3 Kerckhoffs' Prinzip: Nur der Schlüssel wird geheim gehalten	8
1.4 Asymmetrische Kryptografie: Zwei Schlüssel sind besser als einer	10
1.4.1 Schlüsselaustausch oder wie man zu einem gemeinsamen Geheimnis kommt	11
1.4.2 Asymmetrische Verschlüsselung – anders als die symmetrische	14
1.4.3 Digitale Signaturen – wie Unterschrift mit Stift und Papier	16
1.5 Klassifizierende und abstrahierende Kryptografie	18
1.6 Theoretische Kryptografie vs. praktische Kryptografie	20
1.7 Von der Theorie zur Praxis: Wählen Sie Ihr eigenes Abenteuer	21
1.8 Ein Wort der Warnung	27
Zusammenfassung	27

<b>2</b>	<b>Hashfunktionen</b>	<b>29</b>
2.1	Was ist eine Hashfunktion? . . . . .	29
2.2	Sicherheitseigenschaften einer Hashfunktion . . . . .	32
2.3	Sicherheitsbetrachtungen für Hashfunktionen . . . . .	34
2.4	Hashfunktionen in der Praxis . . . . .	36
2.4.1	Commitments . . . . .	36
2.4.2	Subressourcenintegrität . . . . .	37
2.4.3	BitTorrent . . . . .	37
2.4.4	Tor . . . . .	38
2.5	Standardisierte Hashfunktionen . . . . .	39
2.5.1	Die Hashfunktion SHA-2 . . . . .	40
2.5.2	Die Hashfunktion SHA-3 . . . . .	43
2.5.3	SHAKE und cSHAKE: Zwei Funktionen mit erweiterbarer Ausgabe (XOF) . . . . .	48
2.5.4	Mehrdeutiges Hashing mit TupleHash vermeiden . . . . .	49
2.6	Hashing von Kennwörtern . . . . .	51
	Zusammenfassung . . . . .	53
<b>3</b>	<b>Message Authentication Codes (MACs)</b>	<b>55</b>
3.1	Zustandslose Cookies, ein motivierendes Beispiel für MACs . . . . .	55
3.2	Ein Beispiel in Code . . . . .	59
3.3	Sicherheitseigenschaften eines MAC . . . . .	60
3.3.1	Fälschen eines Authentifizierungstags . . . . .	60
3.3.2	Längen des Authentifizierungstags . . . . .	61
3.3.3	Replay-Angriffe . . . . .	62
3.3.4	Authentifizierungstags in konstanter Zeit verifizieren . . . . .	63
3.4	MAC im wahren Leben . . . . .	65
3.4.1	Authentifizierung von Nachrichten . . . . .	65
3.4.2	Schlüssel ableiten . . . . .	65
3.4.3	Integrität von Cookies . . . . .	66
3.4.4	Hashtabellen . . . . .	66
3.5	MACs in der Praxis . . . . .	67
3.5.1	HMAC, ein Hash-basierter MAC . . . . .	67
3.5.2	KMAC – ein MAC, der auf cSHAKE basiert . . . . .	68
3.6	SHA-2- und Length-Extension-Angriffe . . . . .	69
	Zusammenfassung . . . . .	72

<b>4</b>	<b>Authentifizierte Verschlüsselung</b>	<b>73</b>
4.1	Was ist eine Chiffre? . . . . .	74
4.2	Die Blockchiffre AES (Advanced Encryption Standard) . . . . .	76
4.2.1	Wie viel Sicherheit bietet AES? . . . . .	76
4.2.2	Die Schnittstelle von AES . . . . .	77
4.2.3	Die Interna von AES . . . . .	78
4.3	Der verschlüsselte Pinguin und die Betriebsart CBC . . . . .	80
4.4	Fehlende Authentizität, deshalb AES-CBC-HMAC . . . . .	83
4.5	All-in-one-Konstruktionen: Authentifizierte Verschlüsselung . . . . .	85
4.5.1	Was ist authentifizierte Verschlüsselung mit zugehörigen Daten (AEAD)? . . . . .	85
4.5.2	Der AEAD-Modus AES-GCM . . . . .	87
4.5.3	ChaCha20-Poly1305 . . . . .	92
4.6	Andere Arten der symmetrischen Verschlüsselung . . . . .	96
4.6.1	Key-Wrapping . . . . .	97
4.6.2	Authentifizierte Verschlüsselung, die gegen Nonce-Missbrauch resistent ist . . . . .	97
4.6.3	Datenträgerverschlüsselung . . . . .	97
4.6.4	Datenbankverschlüsselung . . . . .	98
	Zusammenfassung . . . . .	99
<b>5</b>	<b>Schlüsselaustausch</b>	<b>101</b>
5.1	Was sind Schlüsselvereinbarungen? . . . . .	102
5.2	Der Diffie-Hellman-(DH-)Schlüsselaustausch . . . . .	105
5.2.1	Gruppentheorie . . . . .	105
5.2.2	Das Problem des diskreten Logarithmus: Die Basis von Diffie-Hellman . . . . .	110
5.2.3	Die Diffie-Hellman-Standards . . . . .	112
5.3	Der Elliptic Curve Diffie-Hellman-(ECDH-)Schlüsselaustausch . . . . .	113
5.3.1	Was ist eine elliptische Kurve? . . . . .	114
5.3.2	Wie funktioniert der Elliptic Curve Diffie-Hellman- (ECDH-)Schlüsselaustausch? . . . . .	118
5.3.3	Die Standards für Elliptic Curve Diffie-Hellman . . . . .	119
5.4	Angriffe auf kleine Untergruppen und andere Sicherheits- überlegungen . . . . .	121
	Zusammenfassung . . . . .	125

<b>6</b>	<b>Asymmetrische und hybride Verschlüsselung</b>	<b>127</b>
6.1	Was ist asymmetrische Verschlüsselung? . . . . .	128
6.2	Asymmetrische Verschlüsselung in der Praxis und hybride Verschlüsselung . . . . .	130
6.2.1	Schlüsselvereinbarungen und Schlüsselkapselung . . . . .	130
6.2.2	Hybride Verschlüsselung . . . . .	132
6.3	Asymmetrische Verschlüsselung mit RSA: Das Schlechte und das weniger Schlechte . . . . .	136
6.3.1	RSA nach Lehrbuch . . . . .	136
6.3.2	Warum man RSA PKCS#1 v1.5 nicht verwenden sollte . . .	141
6.3.3	Asymmetrische Verschlüsselung mit RSA-OAEP . . . . .	142
6.4	Hybride Verschlüsselung mit ECIES . . . . .	146
	Zusammenfassung . . . . .	148
<b>7</b>	<b>Signaturen und Null-Wissen-Beweise</b>	<b>149</b>
7.1	Was ist eine Signatur? . . . . .	150
7.1.1	Signieren und Verifizieren in der Praxis . . . . .	151
7.1.2	DER Anwendungsfall von Signaturen: Authentifizierter Schlüsselaustausch . . . . .	152
7.1.3	Eine praktische Anwendung: Infrastrukturen für öffentliche Schlüssel . . . . .	153
7.2	Null-Wissen-Beweise (ZKPs): Der Ursprung der Signaturen . . . . .	155
7.2.1	Schnorr-Identifikationsprotokoll: Ein interaktiver Null-Wissen-Beweis . . . . .	155
7.2.2	Signaturen als nicht interaktive Null-Wissen-Beweise . . .	159
7.3	Die Signaturalgorithmen, die Sie verwenden sollten (oder nicht) . .	160
7.3.1	RSA PKCS#1 v1.5: Ein schlechter Standard . . . . .	161
7.3.2	RSA-PSS: Ein besserer Standard . . . . .	164
7.3.3	Der Elliptic Curve Digital Signature-Algorithmus (ECDSA) . . . . .	166
7.3.4	Der Edwards-curve Digital Signature Algorithm (EdDSA) . . . . .	168
7.4	Subtiles Verhalten von Signaturverfahren . . . . .	172
7.4.1	Substitutionsangriffe auf Signaturen . . . . .	172
7.4.2	Malleability von Signaturen . . . . .	174
	Zusammenfassung . . . . .	175
<b>8</b>	<b>Zufälligkeit und Geheimnisse</b>	<b>177</b>
8.1	Was ist Zufälligkeit? . . . . .	178
8.2	Langsame Zufälligkeit? Verwenden Sie einen Pseudozufalls- zahlengenerator (PRNG) . . . . .	180
8.3	Zufälligkeit in der Praxis erzeugen . . . . .	184

8.4	Zufallszahlenerzeugung und Sicherheitsüberlegungen . . . . .	186
8.5	Öffentliche Zufälligkeit . . . . .	189
8.6	Schlüsselableitung mit HKDF . . . . .	191
8.7	Schlüssel und Geheimnisse verwalten . . . . .	195
8.8	Dezentralisiertes Vertrauen mit Schwellenwertkryptografie . . . . .	197
	Zusammenfassung . . . . .	200

---

**Teil B Protokolle: Die Rezepte der Kryptografie 203**

---

<b>9</b>	<b>Sicherer Transport</b>	<b>205</b>
9.1	Die Protokolle für sicheren Transport – SSL und TLS . . . . .	205
9.1.1	Von SSL zu TLS . . . . .	206
9.1.2	TLS in der Praxis verwenden . . . . .	207
9.2	Wie funktioniert das TLS-Protokoll? . . . . .	209
9.2.1	Der TLS-Handshake . . . . .	210
9.2.2	Wie TLS 1.3 Anwendungsdaten verschlüsselt . . . . .	224
9.3	Der Stand der Dinge im verschlüsselten Web heute . . . . .	225
9.4	Andere sichere Transportprotokolle . . . . .	228
9.5	Das Noise-Protokoll-Framework: Eine moderne Alternative zu TLS . . . . .	229
9.5.1	Die vielen Handshakes von Noise . . . . .	229
9.5.2	Ein Handshake mit Noise . . . . .	230
	Zusammenfassung . . . . .	232
<b>10</b>	<b>Ende-zu-Ende-Verschlüsselung</b>	<b>233</b>
10.1	Warum Ende-zu-Ende-Verschlüsselung? . . . . .	234
10.2	Eine Vertrauensbasis, die nirgendwo zu finden ist . . . . .	236
10.3	Das Scheitern der verschlüsselten E-Mail . . . . .	237
10.3.1	PGP oder GPG? Und wie funktionieren sie? . . . . .	238
10.3.2	Vertrauen zwischen Benutzern mit dem Netz des Vertrauens skalieren . . . . .	241
10.3.3	Schlüsselermittlung ist ein echtes Problem . . . . .	242
10.3.4	Wenn nicht PGP, was dann? . . . . .	243
10.4	Sicheres Messaging: Ein moderner Blick auf die Ende-zu-Ende- Verschlüsselung mit Signal . . . . .	245
10.4.1	Benutzerfreundlicher als WOT: Vertrauen, aber verifizieren . . . . .	246
10.4.2	X3DH: Der Handshake des Signal-Protokolls . . . . .	249
10.4.3	Double Ratchet: Das Post-Handshake-Protokoll von Signal . . . . .	252
10.5	Der Stand der Ende-zu-Ende-Verschlüsselung . . . . .	257
	Zusammenfassung . . . . .	259

<b>11</b>	<b>Benutzerauthentifizierung</b>	<b>261</b>
11.1	Authentifizierung – eine Wiederholung	262
11.2	Benutzerauthentifizierung – oder wie wird man Kennwörter los? ...	264
11.2.1	Ein Kennwort für alles: Single Sign-on (SSO) und Kennwort-Manager	266
11.2.2	Kein Interesse an Ihren Kennwörtern? Verwenden Sie einen asymmetrischen kennwort- authentifizierten Schlüsselaustausch	268
11.2.3	Einmalkennwörter sind eigentlich keine Kennwörter: Mit symmetrischen Schlüsseln kennwortlos werden	272
11.2.4	Kennwörter durch asymmetrische Schlüssel ersetzen	276
11.3	Benutzergestützte Authentifizierung: Pairing von Geräten mit menschlicher Hilfe	279
11.3.1	Vorher vereinbarte Schlüssel (Pre-shared keys)	281
11.3.2	Symmetrischer kennwortauthentifizierter Schlüssel- austausch mit CPace	283
11.3.3	Gab es einen MITM-Angriff auf meinen Schlüssel- austausch? Prüfen Sie einfach einen kurzen authentifizierten String (SAS)	284
	Zusammenfassung	288
<b>12</b>	<b>Krypto wie in Kryptowährung?</b>	<b>291</b>
12.1	Eine kleine Einführung in byzantinische fehlertolerante (BFT) Konsensalgorithmen	292
12.1.1	Ein Problem der Stabilität: Verteilte Protokolle zur Rettung	292
12.1.2	Ein Problem des Vertrauens? Dezentralisierung hilft	294
12.1.3	Ein Problem der Größe: Erlaubnisfreie und zensur- resistente Netzwerke	295
12.2	Wie funktioniert Bitcoin?	297
12.2.1	Wie Bitcoin mit Kontoständen und Transaktionen umgeht	298
12.2.2	BTCs schürfen im digitalen Goldzeitalter	300
12.2.3	Verzweigungschaos – Konflikte beim Mining lösen	304
12.2.4	Die Blockgröße mit Merkle-Bäumen reduzieren	307
12.3	Ein Rundgang durch die Kryptowährungen	309
12.3.1	Volatilität	309
12.3.2	Latenz	309
12.3.3	Größe der Blockchain	310
12.3.4	Vertraulichkeit	310
12.3.5	Energieeffizienz	310

12.4	DiemBFT: Ein byzantinisch fehlertolerantes (BFT) Konsensprotokoll . . . . .	311
12.4.1	Sicherheit und Lebendigkeit: Die beiden Eigenschaften eines BFT-Konsensprotokolls . . . . .	311
12.4.2	Eine Runde im DiemBFT-Protokoll . . . . .	312
12.4.3	Wie viel Unehrlichkeit kann das Protokoll tolerieren? . . .	313
12.4.4	Die DiemBFT-Regeln für eine Abstimmung . . . . .	314
12.4.5	Wann gelten Transaktionen als finalisiert? . . . . .	315
12.4.6	Die Intuitionen hinter der Sicherheit von DiemBFT . . . . .	316
	Zusammenfassung . . . . .	318
<b>13</b>	<b>Hardware-Kryptografie</b>	<b>321</b>
13.1	Angreifermodell der modernen Kryptografie . . . . .	321
13.2	Nicht vertrauenswürdige Umgebungen: Hardware als Rettung . . .	323
13.2.1	White-Box-Kryptografie – eine schlechte Idee . . . . .	324
13.2.2	In Ihrer Brieftasche: Smartcards und Secure Elements . . .	325
13.2.3	Lieblinge der Banken: Hardware-Sicherheitsmodule (HSMs) . . . . .	328
13.2.4	Trusted Platform Modules (TPMs): Eine nützliche Standardisierung von Secure Elements . . . . .	330
13.2.5	Vertrauliche Datenverarbeitung mit einer vertrauens- würdigen Ausführungsumgebung (TEE) . . . . .	334
13.3	Welche Lösung ist für mich geeignet? . . . . .	335
13.4	Leakage-resiliente Kryptografie oder wie man Seitenkanal- angriffe in Software entschärft . . . . .	337
13.4.1	Programmierung in konstanter Zeit . . . . .	340
13.4.2	Nicht das Geheimnis verwenden! Maskieren und Blinding . . . . .	341
13.4.3	Was ist mit Fehlerangriffen? . . . . .	342
	Zusammenfassung . . . . .	343
<b>14</b>	<b>Post-Quanten-Kryptografie</b>	<b>347</b>
14.1	Was sind Quantencomputer und warum fürchten sich Kryptografen vor ihnen? . . . . .	348
14.1.1	Quantenmechanik – das Studium des Kleinen . . . . .	348
14.1.2	Von der Geburt des Quantencomputers zur Quanten- überlegenheit . . . . .	351
14.1.3	Der Einfluss der Algorithmen von Grover und Shor auf die Kryptografie . . . . .	353
14.1.4	Post-Quanten-Kryptografie – die Verteidigung gegen Quantencomputer . . . . .	354

14.2	Hash-basierte Signaturen – eine Hashfunktion genügt . . . . .	355
14.2.1	Lamport-Einmal-Signaturverfahren . . . . .	355
14.2.2	Kleinere Schlüssel mit Winternitz-Einmal-Signaturen (WOTS) . . . . .	357
14.2.3	Vielfache Signaturen mit XMSS und SPHINCS+ . . . . .	359
14.3	Kürzere Schlüssel und Signaturen mit gitterbasierter Kryptografie . .	362
14.3.1	Was ist ein Gitter? . . . . .	362
14.3.2	Lernen mit Fehlern (LWE), eine Basis für die Kryptografie? . . . . .	364
14.3.3	Kyber, ein gitterbasierter Schlüsselaustausch . . . . .	366
14.3.4	Dilithium, ein gitterbasiertes Signaturverfahren . . . . .	368
14.4	Muss ich in Panik geraten? . . . . .	370
	Zusammenfassung . . . . .	372
<b>15</b>	<b>Ist es das? Die Kryptografie der nächsten Generation</b>	<b>375</b>
15.1	Je mehr, desto besser: Sichere Mehrparteienberechnung (MPC) . . .	376
15.1.1	Private Mengenüberschneidung (PSI) . . . . .	377
15.1.2	MPC für allgemeine Zwecke . . . . .	378
15.1.3	Der Zustand von MPC . . . . .	380
15.2	Vollständig homomorphe Verschlüsselung (FHE) und die Versprechen einer verschlüsselten Cloud . . . . .	381
15.2.1	Ein Beispiel für homomorphe Verschlüsselung mit RSA-Verschlüsselung . . . . .	381
15.2.2	Die verschiedenen Arten der homomorphen Verschlüsselung . . . . .	382
15.2.3	Bootstrapping, der Schlüssel zur vollständig homomorphen Verschlüsselung . . . . .	382
15.2.4	Ein FHE-Schema, das auf dem Problem Lernen mit Fehlern basiert . . . . .	385
15.2.5	Wo wird es verwendet? . . . . .	386
15.3	Allgemeine Null-Wissen-Beweise (ZKPs) . . . . .	387
15.3.1	Wie zk-SNARKs funktionieren . . . . .	390
15.3.2	Homomorphe Commitments, um Teile des Beweises zu verbergen . . . . .	391
15.3.3	Bilineare Paarungen, um unsere homomorphen Commitments zu verbessern . . . . .	392
15.3.4	Woher kommt die Prägnanz? . . . . .	393
15.3.5	Von Programmen zu Polynomen . . . . .	394
15.3.6	Programme sind für Computer; wir brauchen stattdessen arithmetische Schaltungen . . . . .	394

---

15.3.7	Eine arithmetische Schaltung in ein Rang-1-Constraint-System (R1CS) konvertieren . . . . .	395
15.3.8	Von R1CS zu einem Polynom . . . . .	396
15.3.9	Es gehören zwei dazu, um ein im Exponenten verstecktes Polynom auszuwerten . . . . .	397
	Zusammenfassung . . . . .	399
<b>16</b>	<b>Wann und wo Kryptografie scheitert</b>	<b>401</b>
16.1	Die Suche nach dem richtigen kryptografischen Primitiv oder Protokoll ist eine langweilige Angelegenheit . . . . .	402
16.2	Wie verwende ich ein kryptografisches Primitiv oder Protokoll? Höfliche Standards und formale Verifizierung . . . . .	404
16.3	Wo sind die guten Bibliotheken? . . . . .	407
16.4	Kryptografie missbrauchen: Entwickler sind der Feind . . . . .	408
16.5	Sie machen es falsch: Brauchbare Sicherheit . . . . .	410
16.6	Kryptografie ist keine Insel . . . . .	411
16.7	Ihre Verantwortlichkeiten als Kryptografie-Praktiker – keine »selbst gedrehte« Krypto . . . . .	412
	Zusammenfassung . . . . .	414
<b>A</b>	<b>Antworten zu den Übungen</b>	<b>417</b>
A.1	Kapitel 2 . . . . .	417
A.2	Kapitel 3 . . . . .	418
A.3	Kapitel 6 . . . . .	418
A.4	Kapitel 7 . . . . .	419
A.5	Kapitel 8 . . . . .	419
A.6	Kapitel 9 . . . . .	419
A.7	Kapitel 10 . . . . .	420
A.8	Kapitel 11 . . . . .	420
	<b>Index</b>	<b>423</b>