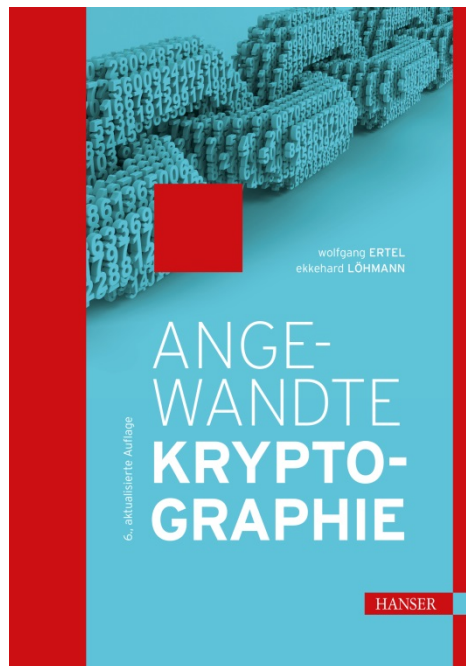


# HANSER



## Leseprobe

zu

## „Angewandte Kryptographie“

von Wolfgang Ertel und Ekkhard Löhmann

Print-ISBN 978-3-446-46313-4

E-Book-ISBN 978-3-446-46353-0

Weitere Informationen und Bestellungen unter  
<http://www.hanser-fachbuch.de/978-3-446-46313-4>

sowie im Buchhandel

© Carl Hanser Verlag, München

# Vorwort

## Ziele

Das Verschlüsseln von Nachrichten oder geheimen Schriftstücken übt auch heute noch eine große Faszination auf Menschen aller Bevölkerungsschichten aus. Die verschiedensten Fachleute aus Mathematik, Informatik und Linguistik beschäftigen sich mit dieser alten Wissenschaft, die bis zur Mitte des zwanzigsten Jahrhunderts hauptsächlich militärisch angewendet wurde.

Dieses Bild hat sich in den letzten dreißig Jahren gewandelt. Im Zeitalter der Globalisierung und des E-Business ist die Welt vernetzt. Heute werden Pläne, Patente, Verträge und andere vertrauliche Daten auf Rechnern gespeichert und über das Internet ausgetauscht. Der rege Datenaustausch weckt großes Interesse bei Geheimdiensten, bei Firmen, die Informationen über ihre Kunden sammeln, sowie bei Firmen, die die Geheimnisse der Konkurrenz ausspionieren wollen. Allein in Deutschland entstehen der Industrie pro Jahr geschätzte Verluste im Bereich zehn bis hundert Milliarden Euro durch Lauschangriffe.

Diese Angriffe geschehen im Stillen und werden in den meisten Fällen der Presse nicht mitgeteilt. Teilweise sind sie sogar der geschädigten Firma nicht bekannt. Oft wird daher die Sicherheit von Firmennetzen gegenüber Angriffen von außen immer noch sträflich vernachlässigt, obwohl Wissen und solide Technik der Datensicherheit heute für jeden Fachmann zugänglich sind. Das wichtigste Ziel des Buches ist es deshalb, dem Informatiker das benötigte Wissen auf einer soliden Basis zu vermitteln. Damit wird er in der Lage sein, zum Beispiel ein Sicherheitskonzept für eine Firma zu erarbeiten oder eine Public-Key-Infrastruktur aufzubauen und zu pflegen.

Es gibt aber auch Beispiele von erfolgreichen Firmen, die plötzlich vor dem Bankrott stehen, nur weil jemand eine gefälschte E-Mail im Namen der Firmenleitung an die Presse schickt, mit der Folge eines dramatischen Absturzes des Aktienkurses. Das Stichwort zur Vermeidung derartiger Fälle heißt digitale Signatur. Die digitale Signatur wird in den nächsten Jahren das Medium E-Mail zu einem seriösen Werkzeug machen, mit dem Verträge, Angebote, Rechnungen etc. schnell, kostengünstig und sicher abgewickelt werden können. Seit Ende 2010 gibt es in Deutschland den neuen Personalausweis mit Chipkarte, der auch für die digitale Signatur benutzt werden kann. Möglich wäre auch die Verwendung des Personalausweises als Schlüssel zu Wohnung, Firma, Rechner und Auto.

Offene Systeme und weltweite Vernetzung führen auch zu Ängsten und zum Wunsch nach Sicherheit, Vertraulichkeit und einem besseren Schutz der Privatsphäre. Sicher ist es kein Zufall, dass gerade zum jetzigen Zeitpunkt mit der vor gut zwanzig Jahren erfundenen

Public-Key-Kryptographie und den modernen Blockchiffren starke und mittlerweile bewährte Werkzeuge zur Sicherung der Privatsphäre und Vertraulichkeit zum Einsatz in der Praxis bereitstehen. Ziel dieses Buches ist es, den Leser mit diesen Methoden vertraut zu machen und zwar ausgehend von den teilweise genial einfachen und eleganten Ideen über die Mathematik endlicher Körper bis hin zu den Anwendungen in Form von allgemein verfügbarer Software.

Die Aussage „mein Computer ist sicher“ ist eine All-Aussage, denn etwas genauer formuliert heißt sie „die Erfolgswahrscheinlichkeit für einen der vielen möglichen Angriffe ist verschwindend gering“. Um solch eine Aussage auch nur annähernd machen zu können, muss jede Schwachstelle beseitigt werden, denn ein kluger Angreifer nutzt die schwächste Stelle – und die Tücken liegen im Detail. Nur durch den praktischen Umgang mit der Materie ist es möglich, aufbauend auf den theoretischen Grundlagen, die benötigte umfassende Vorgehensweise zur Aufdeckung und Beseitigung von Sicherheitslücken zu erlernen. Das Wissen über die Algorithmen und die Mathematik von Kryptosystemen ist notwendig, aber bei weitem nicht hinreichend, um sichere Systeme zu bauen. Daher möchte ich den motivierten Neuling in diesem Gebiet insbesondere auffordern, die Übungsaufgaben zu bearbeiten.

### **Aufbau und Leserkreis**

Das Buch ist entstanden aus einem Vorlesungsskript zur Datensicherheit im Informatikstudium an der Fachhochschule Ravensburg-Weingarten. Es ist ein Lehrbuch zur Einführung in das Gebiet und richtet sich primär an Studenten der Fachhochschulen, aber auch an Universitätsstudenten, die sich ohne viel Theorie in das Gebiet einarbeiten wollen. Wie man schon am Titel erkennt, habe ich versucht, die Theorie auf ein Minimum zu beschränken. Das Buch wendet sich deshalb an alle, die in kompakter Form die moderne Kryptographie verstehen wollen. Dem berufstätigen Informatiker bietet es die Möglichkeit, sich im Selbststudium in ein aktuelles Gebiet einzuarbeiten.

Vorausgesetzt werden Mathematikkenntnisse der Oberstufe. Darüber hinaus benötigte Mathematik wird im Anhang A bereitgestellt. Das Buch beginnt mit einer elementaren Einführung in die Protokolle für elektronisches Bargeld als Beispiel einer Anwendung für viele im Buch beschriebene Algorithmen und Protokolle. Nach den Grundlagen in Kapitel 2 werden im Kapitel 3 an Hand einiger klassischer Chiffren wichtige Techniken und Begriffe eingeführt.

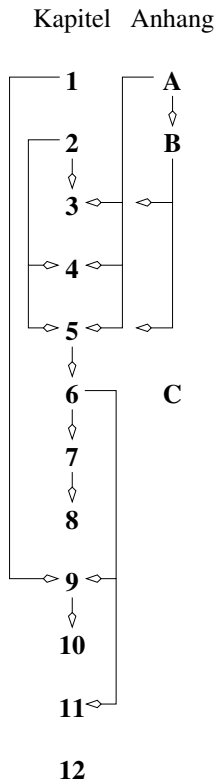
Bei den modernen Blockchiffren in Kapitel 4 werden DES, die weltweit meist benutzte Chiffre, und AES als neuer Standard vorgestellt. Die Public-Key-Kryptographie ist in den Kapiteln 5, 7 und 8 behandelt und es wird neben den Algorithmen ausführlich auf die Public-Key-Infrastruktur sowie auf die wichtigsten Software-Produkte eingegangen. Aufbauend auf den Public-Key-Algorithmen werden in Kapitel 6 neben klassischen Authentifikationsverfahren die digitale Signatur sowie Zero-Knowledge-Protokolle behandelt.

Nachdem alle Techniken eingeführt sind, schließt sich der Kreis und die Protokolle für elektronisches Bargeld aus Kapitel 1 werden in Kapitel 9 verfeinert und exakt beschrieben. Kapitel 10 schließlich stellt verschiedene existierende und neue elektronische Zahlungsmittel vor und vergleicht sie.

In Kapitel 12 wird das deutsche Signaturgesetz vorgestellt sowie das politische und gesellschaftliche Umfeld der modernen Kryptographie beleuchtet. Als Abschluss folgt in Kapi-

tel 13 eine Checkliste für die praktische Arbeit in der Kryptographie. Die benötigte Zahlentheorie, ein Kapitel über die Erzeugung von Zufallszahlen für kryptographische Algorithmen und die Lösungen zu den Übungsaufgaben sind im Anhang zu finden.

Die Abhängigkeit der Kapitel untereinander ist in Bild 1 dargestellt. Ein Pfeil von 2 nach 3 zum Beispiel bedeutet, dass Kapitel 2 für das Verständnis von Kapitel 3 vorausgesetzt wird.



**BILD 1** Kapitelstruktur

der ein gutes Nachschlagewerk sucht, findet dieses in Form des umfassenden und sehr gut lesbaren Standardwerkes von Bruce Schneier [Sch05, Sch96]. Empfehlenswerte Lehrbücher sind [Sti05, Kob94, Sta98, Wob01, Beu09, Bau00].

## Dank

Mein ganz besonderer Dank gilt meiner Frau Evelyn, die mir im letzten Jahr den Rücken frei hielt für das Schreiben. Vielen Dank auch an Ekkehard Löhmann für wertvolle inhaltliche Tipps und an Erhard Schreck für die schöne Zeit im Silicon Valley, in der das Kapitel über Zufallszahlen entstanden ist. Mein Dank richtet sich auch an Max Kliche für das Bereitstellen der Übungsaufgaben im Web und an Thomas Degen und Ulrich Hauser, die mich

Ich möchte den Leser bitten, Anregungen, Kritik und Hinweise auf Fehler per E-Mail direkt an ertel@hs-weingarten.de zu schicken. Eine regelmäßig aktualisierte Liste der Fehler ist auf der Webseite zum Buch zu finden.

## Online-Quellen und Literatur

Die Web-Seite zum Buch hat die URL

[www.hs-weingarten.de/~ertel/kryptobuch.html](http://www.hs-weingarten.de/~ertel/kryptobuch.html)

Das im Buch abgedruckte Literaturverzeichnis ist dort mit anklickbaren Links versehen, so dass der Leser auf alle im Internet verfügbaren Quellen einfach zugreifen kann. Außerdem gibt es dort eine regelmäßig aktualisierte und nach Themen geordnete Sammlung von Links zur Kryptographie. Ergänzt wird die Sammlung durch Präsentationsfolien für Dozenten.

Neben diesen Quellen möchte ich den interessierten Leser verweisen auf die Newsgroup `sci.crypt`. In diesem stark frequentierten Forum werden die verschiedensten mehr oder weniger aktuellen Themen diskutiert. Sehr informativ sind auch der monatlich erscheinende kostenlose Newsletter „crypto-gram“ von Bruce Schneier [Sch01a], sein neues Buch [Sch00a], sowie die umfangreiche Sammlung von Wissen, Literatur und Links zur Kryptographie von Terry Ritter [Rit00]. Zum praktischen Üben ist das frei verfügbare Demonstrationsprogramm `CrypTool` [Ess02] sehr zu empfehlen.

Es gibt, insbesondere in der englischsprachigen Literatur, eine Reihe guter Lehrbücher zur Kryptographie. Der Leser,

regelmäßig mit aktuellen Schlagzeilen aus den Online-Medien versorgen. Für das Korrekturlesen möchte ich mich bedanken bei Daniel Hirscher, Markus König, Michael König, Norbert Perk und Harald Steinhilber. Meinem Kollegen Martin Hulin danke ich dafür, dass ich mich in den Semesterferien, frei von administrativen Nebenjobs, auf das Schreiben konzentrieren konnte. Bei meiner Lektorin Erika Hotho bedanke ich mich herzlich für die sehr gute Zusammenarbeit.

Ravensburg, den 28. März 2001

Wolfgang Ertel

### **Vorwort zur fünften Auflage**

Neben einigen korrigierten Fehlern wurde ein neues Kapitel über die derzeit viel diskutierte Blockchaintechnologie in das Buch integriert. Dieses wurde verfasst von Ekkehard Löhm, Informatikprofessor an der Hochschule Ravensburg-Weingarten mit langer Berufs- und Lehrerfahrung in der Kryptographie. Die Blockchaintechnologie kommt nicht nur bei der Kryptowährung Bitcoin zum Einsatz, sondern könnte in der Zukunft auch interessant werden zum Erstellen und elektronischen Verwalten von Verträgen.

Ravensburg, den 09. Juli 2018

Wolfgang Ertel

# Inhalt

<b>1</b>	<b>Elektronisches Bargeld, ein erstes Beispiel .....</b>	<b>15</b>
<b>2</b>	<b>Grundlagen .....</b>	<b>21</b>
2.1	Terminologie .....	21
2.2	Kryptographische Algorithmen .....	22
2.3	Kryptographische Protokolle .....	24
2.4	Public-Key-Algorithmen .....	24
2.5	Kryptanalyse .....	26
2.6	Sicherheit von Schlüsseln .....	27
<b>3</b>	<b>Klassische Chiffren .....</b>	<b>31</b>
3.1	Verschiebechiffren .....	32
3.2	Multiplikative Chiffren .....	33
3.3	Tauschchiffren (Affine Chiffren) .....	35
3.4	Kryptanalyse monoalphabetischer Chiffren .....	36
3.5	Polyalphabetische Chiffren .....	37
3.5.1	Homophone Chiffren .....	37
3.6	Die Vigenère-Chiffre .....	38
3.6.1	Der Algorithmus .....	38
3.6.2	Kryptanalyse .....	40
3.6.3	Der Kasiski-Test .....	40
3.6.4	Der Friedman-Test .....	43
3.7	Die Enigma .....	45
3.7.1	Kryptanalyse .....	48
3.8	Das One-Time-Pad, die perfekte Chiffre .....	52
3.9	One-Time-Pad fast ohne Schlüsseltausch .....	55
3.10	Zusammenfassung .....	57

<b>4</b>	<b>Moderne Blockchiffren</b> .....	<b>59</b>
4.1	Data-Encryption-Standard DES .....	59
4.1.1	Übersicht .....	61
4.1.2	Eine Runde .....	63
4.1.3	Die 16 Teilschlüssel .....	64
4.1.4	Die Dechiffrierfunktion .....	64
4.1.5	Sicherheit und Nichtlinearität .....	66
4.1.6	Sicherheit und Geschwindigkeit .....	68
4.1.7	Triple-DES .....	68
4.2	Advanced-Encryption-Standard AES .....	68
4.2.1	Die Blockchiffre Rijndael .....	69
4.2.2	Die ByteSub-Transformation .....	70
4.2.3	Die ShiftRow-Transformation .....	71
4.2.4	Die MixColumn-Transformation .....	72
4.2.5	Die Schlüsselexpansion .....	72
4.2.6	Die inverse Chiffre .....	73
4.2.7	Geschwindigkeit .....	73
4.2.8	Sicherheit .....	73
4.2.9	Andere Funktionalitäten .....	74
4.3	Betriebsmodi von Blockchiffren .....	74
4.4	Andere Blockchiffren .....	75
<b>5</b>	<b>Public-Key-Kryptographie</b> .....	<b>77</b>
5.1	Merkles Rätsel .....	78
5.2	Der RSA-Algorithmus .....	79
5.2.1	Der Algorithmus .....	80
5.2.2	Sicherheit von RSA .....	82
5.2.3	Effiziente Primzahltests .....	83
5.2.4	Effizienz und Implementierung von RSA .....	84
5.2.5	Schnellere Implementierung von RSA .....	85
5.2.6	Angriffe gegen RSA .....	86
5.3	Angriffe gegen Public-Key-Verfahren .....	87
5.3.1	Chosen-Ciphertext-Angriff mit Social Engineering .....	87
5.3.2	Angriffe aufgrund von Seiteneffekten .....	87
5.3.3	Angriffe mit Spezialhardware .....	89
5.4	Schlüsseltausch .....	89
5.4.1	Schlüsseltausch mit symmetrischen Verfahren .....	89
5.4.2	Man-in-the-Middle-Angriff .....	90

5.4.3	Das Interlock-Protokoll .....	90
5.4.4	Schlüsseltausch mit Quantenkryptographie .....	91
5.5	Der Diffie-Hellman-Algorithmus .....	91
5.6	Der ElGamal-Algorithmus .....	93
5.7	Algorithmen mit Elliptischen Kurven .....	93
<b>6</b>	<b>Authentifikation und digitale Signatur .....</b>	<b>97</b>
6.1	Einwegfunktionen und Einweg-Hash-Funktionen .....	98
6.1.1	Passwortverschlüsselung .....	100
6.1.2	Der Geburtstagsangriff .....	100
6.2	Zero-Knowledge-Protokolle .....	102
6.2.1	Challenge-and-Response .....	102
6.2.2	Die Idee der Zero-Knowledge-Protokolle .....	103
6.2.3	Das Fiat-Shamir-Protokoll .....	104
6.3	Digitale Signaturen .....	105
6.3.1	Digital Signature Algorithm (DSA) .....	106
6.3.2	Blinde Signaturen .....	107
6.4	Digitale Signatur in der Praxis .....	108
6.4.1	Speichern des geheimen Schlüssels .....	108
6.4.2	Vertrauen in die Software .....	109
6.4.3	Zusammenfassung .....	110
6.5	Das Signaturgesetz .....	111
6.6	Authentifikation mit digitaler Signatur .....	112
6.7	Message-Authentication-Code (MAC) .....	113
6.8	Biometrische Verfahren .....	114
<b>7</b>	<b>Public-Key-Infrastruktur .....</b>	<b>117</b>
7.1	Persönliche Prüfung öffentlicher Schlüssel .....	117
7.2	Trustcenter .....	118
7.3	Zertifikathierarchie .....	119
7.4	Web-of-Trust .....	120
7.5	Zukunft .....	121
<b>8</b>	<b>Public-Key-Systeme .....</b>	<b>123</b>
8.1	PGP .....	123
8.1.1	Schlüsseltausch mit PGP .....	126
8.1.2	Die Big-Brother-Funktion .....	126
8.1.3	GnuPG .....	127
8.1.4	Angriffe gegen PGP .....	128



8.2	S/MIME und das X.509-Protokoll.....	130
8.3	OpenPGP versus S/MIME .....	131
8.4	Secure shell (SSH).....	131
8.5	Secure socket layer (SSL).....	132
8.6	Virtual Private Networking und IP Security .....	133
8.7	Der neue Personalausweis.....	134
8.7.1	Hoheitliche Funktionen .....	134
8.7.2	Andere Funktionen .....	135
8.7.3	Digitale Signatur.....	135
8.7.4	Sicherheit des neuen Personalausweises .....	136
<b>9</b>	<b>Elektronisches Bargeld .....</b>	<b>139</b>
9.1	Secret-Splitting .....	139
9.2	Bit-Commitment-Protokolle .....	140
9.3	Protokolle für elektronisches Bargeld .....	141
<b>10</b>	<b>Elektronische Zahlungssysteme.....</b>	<b>145</b>
10.1	Die Geldkarte .....	146
10.2	Mondex.....	147
10.3	Ecash .....	148
10.4	Zahlung per Kreditkarte.....	148
10.4.1	Secure Electronic Transactions (SET) .....	148
10.4.2	PayPal .....	149
10.4.3	Andere Systeme.....	150
10.5	Zusammenfassung .....	150
<b>11</b>	<b>Blockchain-Technologie und Bitcoin.....</b>	<b>151</b>
11.1	Ein einführendes Beispiel.....	151
11.2	Vom virtuellen verteilten Kassenbuch zu Bitcoin .....	153
11.3	Authentizität der Nachricht .....	153
11.4	Berechnung des Kontostandes.....	154
11.5	Bestätigung der Zahlung durch die Mehrheit der Teilnehmer.....	154
11.6	Worin besteht das mathematische Rätsel? .....	155
11.7	Was sind die Eingabedaten in die Hash-Funktion? .....	156
11.8	Die Blockchain .....	156
11.9	Wie sieht ein Block bei Bitcoin aus? .....	158
11.10	Der Blockheader .....	158
11.11	Wie wird die Gültigkeit eines Blocks überprüft? .....	159
11.12	Die Arbeit eines Miners: Proof of Work (PoW) .....	159

11.13	Steuerung der Höhe des Schwellwertes .....	160
11.14	Das Rennen um die längste Kette.....	161
11.15	Die Geschichte von Bitcoin.....	161
11.16	Gibt es bei der Kryptowährung Bitcoin eine Inflation? .....	162
11.17	Ökologische Aspekte des Bitcoin-Systems (Stromverbrauch) .....	162
11.18	Public-Key-Infrastruktur versus Blockchain .....	164
<b>12</b>	<b>Politische Randbedingungen .....</b>	<b>167</b>
12.1	Starke Kryptographie und der Lauschangriff .....	167
12.2	US-Exportgesetze .....	169
<b>13</b>	<b>Sicherheitslücken in der Praxis .....</b>	<b>171</b>
	<b>Anhang .....</b>	<b>175</b>
<b>A</b>	<b>Arithmetik auf endlichen Mengen .....</b>	<b>175</b>
A.1	Modulare Arithmetik .....	175
A.2	Invertierbarkeit in $\mathbb{Z}_n$ .....	178
A.3	Der Euklidische Algorithmus.....	180
A.4	Die Eulersche $\varphi$ -Funktion .....	183
A.5	Primzahlen.....	185
	A.5.1 Primzahltests .....	186
A.6	Der endliche Körper $GF(2^8)$ .....	190
	A.6.1 Addition .....	190
	A.6.2 Multiplikation.....	190
	A.6.3 Polynome mit Koeffizienten in $GF(2^8)$ .....	191
<b>B</b>	<b>Erzeugen von Zufallszahlen .....</b>	<b>195</b>
B.1	Pseudozufallszahlengeneratoren .....	197
	B.1.1 Lineare Schieberegister mit Rückkopplung .....	198
	B.1.2 Stromchiffren .....	200
B.2	Echte Zufallszahlen.....	201
	B.2.1 Der Neumann-Filter .....	201
B.3	Zusammenfassung .....	203
<b>C</b>	<b>Lösungen zu den Übungen .....</b>	<b>205</b>
	<b>Literatur .....</b>	<b>227</b>
	<b>Index.....</b>	<b>235</b>

# 1

## Elektronisches Bargeld, ein erstes Beispiel

Auch im Zeitalter des bargeldlosen Bezahls besitzt das klassische Bargeld durchaus noch seine Berechtigung. Es ermöglicht eine einfache, schnelle, unverbindliche und kostengünstige Abwicklung des Bezahlvorgangs. Bei hohen Beträgen wird Bargeld wegen des Verlust- und Diebstahlrisikos selten verwendet. Hier bietet der bargeldlose Zahlungsverkehr klare Vorteile. Wegen der Abwicklung über eine Bank oder ein Kreditkarteninstitut und der damit verbundenen Dokumentation kann ein derartiger Bezahlvorgang später geprüft und rekonstruiert werden, z. B. anhand eines Kontoauszuges.

Eine neue Problemstellung ergibt sich im Electronic Commerce, das heißt beim Bezahlen von Waren, Dienstleistungen oder Informationen, die im Internet angeboten werden. Die Kosten für viele dieser Dienste bewegen sich im Bereich von wenigen Cent (Micro-Payment). Daher ist eine bargeldlose Transaktion wie zum Beispiel eine Überweisung oder die Belastung einer Kreditkarte unrentabel. Auch möchte der Kunde für die einmalige oder seltene Nutzung eines Dienstes eventuell keine persönlichen Daten oder Kontodaten angeben. Hierzu bietet sich das Bezahlen mit elektronischen Münzen an. Der Bezahlvorgang besteht nur aus dem Übertragen von einigen elektronischen Münzen, das heißt Bitfolgen zwischen Kunde und Händler. Wie beim klassischen Bargeld werden zwischen den beiden Partnern Objekte – nämlich elektronische Münzen – ausgetauscht. Gegebenenfalls wird auch Wechselgeld zurückgegeben, allerdings werden Kunde und Händler damit nicht belastet. Wie beim klassischen Bargeld sollte das Bezahlen anonym erfolgen, gleichzeitig aber sicher gegen Betrug sein.

Das Bezahlen mit elektronischen Münzen effizient und sicher zu gestalten, ist eine Aufgabe der modernen Kryptographie. Anhand einiger einfacher Ideen soll nun exemplarisch gezeigt werden, wie die im Buch beschriebenen kryptographischen Protokolle und Algorithmen hierzu verwendet werden. Die technischen Details folgen dann in Kapitel 9, wenn die Voraussetzungen dafür geschaffen sind. Bevor wir uns jedoch auf den faszinierenden, nicht immer ganz einfachen Weg zum Verständnis dieser Techniken machen, wollen wir am Beispiel des elektronischen Bargeldes ohne Theorie einen ersten Eindruck von den teilweise genialen Protokollen und der Mächtigkeit der modernen Kryptographie vermitteln.

Wir werden schrittweise ein Protokoll mit interessanten Eigenschaften vorstellen. Es wurde von David Chaum, dem Gründer der holländischen Firma Digicash entwickelt [Cha85, Cha92] und patentiert.

Die an dem Verfahren beteiligte Bank nennen wir E-Bank und als Zahlungsmittel werden E-Münzen benutzt. Eine solche E-Münze besteht letztlich aus einer (endlichen) Folge von

Bytes, analog zu einem Geldschein, der ein spezielles Stück bedrucktes Papier darstellt. Wir versuchen's zuerst mal ganz naiv:

### Protokoll Nr. 1

Die E-Bank erzeugt auf ihrem PC eine Datei mit dem Inhalt: „E-Münze, Wert: 5 €“, wie in Bild 1.1 dargestellt. Dies führt natürlich sofort zur Inflation, wenn die Kunden den Betrag ihrer E-Münzen beliebig ändern.

### Protokoll Nr. 2

Wenn die E-Bank jedoch die E-Münze mit einer Unterschrift versieht, die nur sie und kein anderer erstellen kann, so kann der Kunde, der die Münze auf seinem Rechner speichert, den Betrag nicht mehr abändern. Falls er das versucht, wird die digitale Signatur der Bank ungültig.<sup>1</sup> Er kann jedoch immer noch betrügen, indem er einfach beliebig viele Kopien der E-Münze erzeugt (Bild 1.1). Dies wird verhindert durch Protokoll Nr. 3.

### Protokoll Nr. 3

Wie in Bild 1.1 dargestellt, vergibt die Bank nun für jede Münze eine eindeutige Seriennummer und signiert den gesamten Text bestehend aus Betrag und Seriennummer<sup>2</sup>. Versucht nun jemand, Kopien einer derartigen Münze herzustellen, so wird der Betrug erkannt. Die E-Bank protokolliert nämlich in einer zentralen Datenbank alle eingegangenen Seriennummern und sobald mindestens zwei Münzen mit der gleichen Seriennummer zur E-Bank zurückkommen werden Hausdetektiv und Staatsanwalt benachrichtigt.

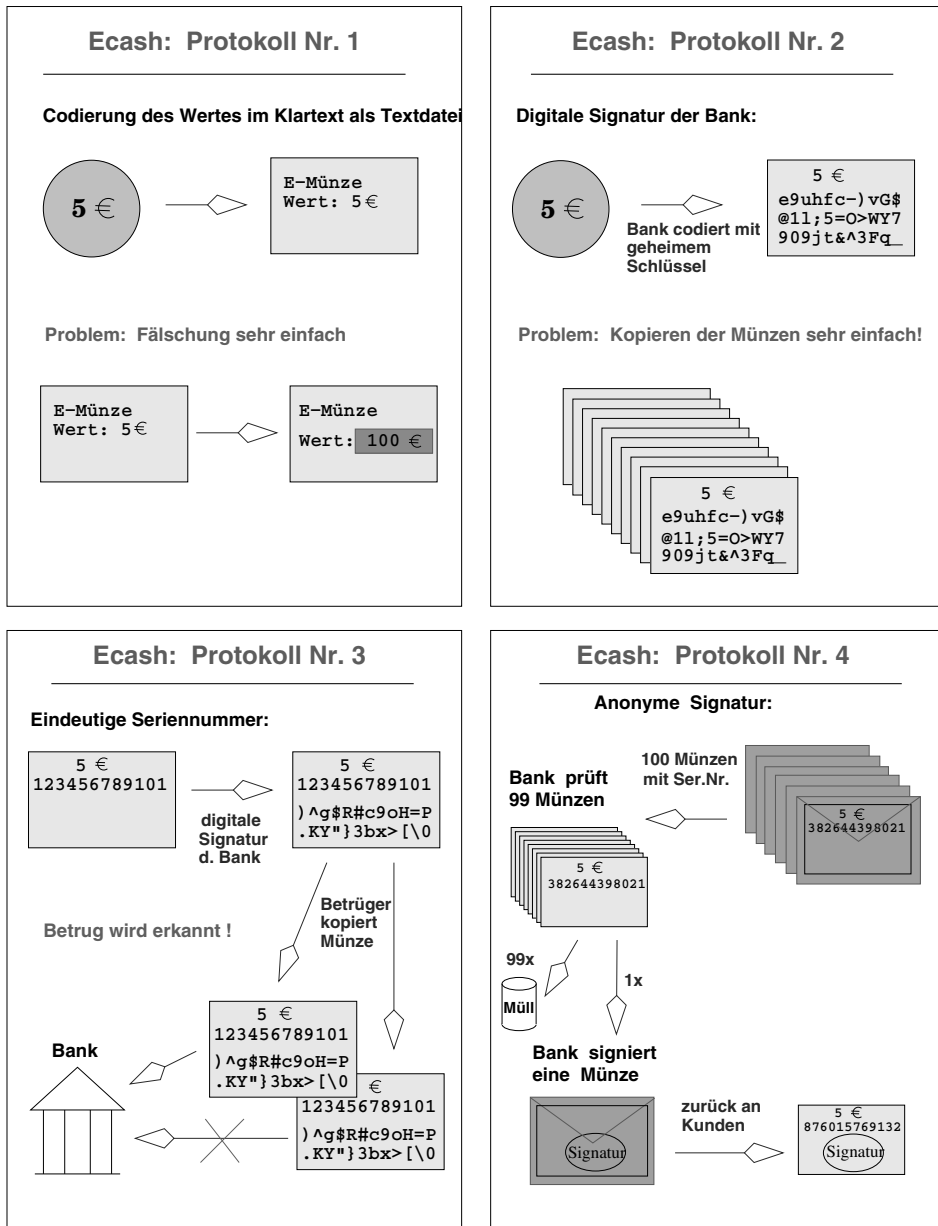
Dieses Protokoll ist sicher, denn jeder Betrug wird erkannt. Es hat aber noch eine Schwäche. Die Anonymität ist nicht gewährleistet, denn die Bank kann aufgrund der Seriennummern ein perfektes Profil jedes Kunden erstellen (siehe Bild 1.2). Das Problem wird offensichtlich durch die Seriennummern verursacht, auf die wir jedoch aus Sicherheitsgründen nicht verzichten können.

### Protokoll Nr. 4

Den Ausweg aus dem Dilemma lieferte David Chaum [Cha85] mit den von ihm erfundenen *blinden Signaturen*. Wie in Bild 1.1 dargestellt, erzeugt nun der Kunde seine E-Münzen selbst. Um eine gültige 5-€-E-Münze zu erhalten, generiert sein PC hundert Dateien, in die jeweils der Text „5 €“ sowie eine große zufällig erzeugte Seriennummer geschrieben werden. Die Seriennummer muss so groß sein, dass die Wahrscheinlichkeit für das zufällige Erzeugen von zwei gleichen Nummern (weltweit) sehr klein ist. Nun bittet er die Bank, eine dieser hundert Münzen blind, das heißt ohne Erkennen von Betrag und Seriennummer, zu signieren. Die Bank wird natürlich nur dann blind signieren, wenn sie sicher ist, dass der Betrag auf der Münze wirklich 5 € ist. Daher wählt sie zufällig 99 der 100 Münzen,

<sup>1</sup> Dies ist ganz analog zu einem unterschriebenen Vertrag, der nicht mehr abgeändert werden darf. Bei digitalen Unterschriften ist das Ändern jedoch nicht mehr möglich.

<sup>2</sup> In realen Implementierungen wird die Bank weitere Informationen, wie z. B. den Namen der Bank und das Datum, auf der E-Münze speichern. Wir beschränken uns hier jedoch auf die zum Verständnis wesentlichen Daten.



Seriennummer	Ausgabe	Kunde	Kto.-Nr.	Händler	Rücklauf	Betrag
123456789101	12.2.2001	Maier	7654321	Otto Versand	14.2.2001	50 €
123456789102	12.2.2001	Maier	7654321	Otto Versand	14.2.2001	20 €
123456789103	12.2.2001	Maier	7654321	Otto Versand	14.2.2001	8 €
123456789104	12.2.2001	Maier	7654321	Otto Versand	14.2.2001	0.90 €
123456789105	12.2.2001	Maier	7654321	amazon.de	17.2.2001	20 €
123456789106	12.2.2001	Maier	7654321	amazon.de	17.2.2001	2 €
123456789107	15.2.2001	Huber	0054322	Frisör Kurz	15.2.2001	20 €
123456789108	15.2.2001	Huber	0054322	Frisör Kurz	15.2.2001	20 €
123456789109	15.2.2001	Huber	0054322	Frisör Kurz	15.2.2001	5 €
123456789110	15.2.2001	Huber	0054322	Frisör Kurz	15.2.2001	1 €
123456789111	15.2.2001	Huber	0054322	Tankst. Sprit	16.2.2001	100 €
123456789112	15.2.2001	Huber	0054322	Tankst. Sprit	16.2.2001	2 €
123456789113	15.2.2001	Huber	0054322	Tankst. Sprit	16.2.2001	2 €
⋮	⋮	⋮	⋮	⋮	⋮	⋮

**BILD 1.2** Beispiel einer möglichen Datenbank von Transaktionen der Kunden der E-Bank

Der Kunde erstellt also 100 Fünfeuroscheine mit Betrag und Seriennummer, packt jeden in einen eigenen Umschlag und legt in den Umschlag über den Geldschein ein Kohlepapier. Die Bank signiert nun den von ihr ausgewählten Geldschein blind, indem sie ihren Stempel aus dem Tresor holt und den Geldschein durch den Umschlag stempelt. Das Kohlepapier hinterlässt auf dem Schein dann den Stempelabdruck. Der Kunde erhält den signierten (gestempelten) Geldschein zurück, packt ihn aus und kann nun damit einkaufen gehen, ohne dass die Bank eine Chance hat, seine Einkäufe zu überwachen. Der Kunde oder auch der Händler kann versuchen, die gültige Münze zu kopieren. Die Bank wird jedoch den Betrug erkennen, weil sie die Seriennummern aller eingehenden Münzen mit den schon eingegangenen in ihrer Datenbank vergleicht. Das Protokoll ist nun also anonym und sicher zugleich.

Ein kleines Problem bleibt jedoch noch zu lösen. Versucht nämlich der Kunde oder der Händler Betrug durch Kopieren der E-Münze, so weiß die Bank zwar, dass der Betrug versucht wurde. Sie weiß jedoch nicht, wer der Betrüger war. David Chaum hat aber auch dieses Problem durch eine elegante Verfeinerung des Protokolls gelöst, die jedoch erst in Kapitel 9 beschrieben werden kann. Hier sei nur so viel verraten: Kopiert der Kunde den Geldschein, so legt die Bank beide eingegangenen Geldscheine übereinander, hält sie gegen das Licht und kann nun den Namen des Betrügers lesen. Ein Geldschein alleine verrät jedoch nichts über die Identität seines Erzeugers.



## Übungen

### Aufgabe 1.1

- a) Ein Betrüger möchte eine Bank, die Protokoll Nr. 4 benutzt, dazu bringen, blind eine 100-€-Münze zu signieren, seinem Konto aber nur einen Euro zu belasten.

Dazu erzeugt er 99 Münzen vom Wert 1 € und eine 100-€-Münze. Wie groß ist die Wahrscheinlichkeit dafür, dass die Bank blind die 100-€-Münze signiert?

b) Wie kann die Bank verhindern, dass der Kunde einen Betrugsversuch unternimmt?

### **Aufgabe 1.2**

Wie viele Bit muss die zufällig generierte Seriennummer einer E-Münze lang sein, damit die Wahrscheinlichkeit für eine zufällige Übereinstimmung von zwei Nummern kleiner ist als die Wahrscheinlichkeit, bei zwei aufeinander folgenden Ziehungen im Lotto (6 aus 49) sechs Richtige zu tippen? Tipp: Berechnen Sie zuerst die Wahrscheinlichkeit, mit einer zufällig erzeugten Seriennummer eine vorgegebene Zahl fester Länge zu treffen. Bestimmen Sie dann deren Länge  $n$ . In Abschnitt 6.1.2 wird gezeigt, dass die Seriennummer doppelt so lang (d. h.  $2n$ ) sein muss, um eine gleich geringe Wahrscheinlichkeit für eine zufällige Übereinstimmung von zwei beliebigen Nummern zu erreichen. ■





# 2

## Grundlagen

### ■ 2.1 Terminologie

Wie jede Wissenschaft besitzt auch die Kryptographie eine eigene Sprache, deren wichtigste Vokabeln hier kurz vorgestellt werden. Die Begriffe Kryptographie und Kryptologie werden in der Literatur unterschiedlich definiert. Am gebräuchlichsten ist folgende Einteilung: **Kryptographie** wird verstanden als die Lehre der Absicherung von Nachrichten durch Verschlüsseln. **Kryptanalyse** ist die Kunst, Chiffretext aufzubrechen, d. h. den Klartext zu reproduzieren, ohne Kenntnis des Schlüssels. **Kryptologie** vereint Kryptographie und Kryptanalyse.

Bei der **Steganographie** werden geheime Nachrichten nicht verschlüsselt, sondern versteckt. Historisches Beispiel hierfür sind unsichtbare Geheimtinten, die später durch Erwärmen sichtbar gemacht werden können. Heute werden digitale Daten in den niederwertigen Bits der Farbinformation von digitalen Bildern versteckt. Auch Audiodateien eignen sich aufgrund ihres Rauschens für die Steganographie. Wegen der geringen praktischen Bedeutung wird hier nicht auf die verwendeten Techniken eingegangen.

Ein **Alphabet**  $A$  ist eine endliche Menge von Zeichen.  $n = |A|$  ist die Mächtigkeit des Alphabets. Der lesbare Text einer Nachricht (message) wird **Klartext** (plaintext) genannt und mit  $M$  bezeichnet. Er wird als Zeichenkette über dem Alphabet  $A$  gebildet. Zum Beispiel sind  $aaa$  und  $abcabbb$  Klartexte über  $\{a, b, c\}$ . **Geheimtexte** oder **Chiffretexte** sind Zeichenketten über dem gleichen Alphabet  $A$  oder einem anderen Alphabet. Auch die **Schlüssel** sind Zeichenketten.

**Verschlüsselung** oder Chiffrierung bezeichnet das Verfahren, um eine Nachricht unverständlich zu machen. Die **Chiffre**  $E$  (encryption) ist eine invertierbare, d. h. eine umkehrbare Abbildung, welche aus dem Klartext  $M$  und einem Schlüssel  $K$  den Geheimtext  $C$  (ciphertext) erzeugt. Voraussetzung für die Umkehrbarkeit einer Abbildung ist die Injektivität<sup>1</sup>. Die Umkehrung von  $E$  zur Wiederherstellung des Klartextes wird **Entschlüsselung** genannt und mit  $D$  (decryption) bezeichnet.

Entsprechend dieser Definitionen gilt  $E(M) = C$  und  $D(C) = M$ , woraus

$$D(E(M)) = M$$

folgt, denn nach dem Entschlüsseln eines Chiffretextes sollte der Klartext zum Vorschein kommen. Praktisch alle kryptographischen Verfahren haben die Aufgabe, eine der folgenden vier Eigenschaften von Nachrichten zu gewährleisten.

<sup>1</sup> Eine Abbildung  $f: D \rightarrow B$  heißt injektiv, wenn für jedes Paar  $x_1, x_2 \in D$  gilt:  $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$ , d. h. zwei verschiedene Zahlen werden durch  $f$  nie auf den gleichen Wert abgebildet.

**Geheimhaltung:** Ziel der Geheimhaltung ist es, das Lesen einer Nachricht für Unbefugte unmöglich bzw. schwierig zu machen.

**Authentifizierung** oder Authentifikation: Identitätsbeweis des Senders einer Nachricht gegenüber dem Empfänger, d. h. der Empfänger kann sicher sein, dass die Nachricht nicht von einem anderen (unbefugten) Absender stammt.

**Integrität:** Die Nachricht darf während der Übermittlung nicht (von Unbefugten) verändert werden. Sie bewahrt ihre Integrität, das heißt ihre Unverletztheit.

**Verbindlichkeit:** Der Sender kann später nicht leugnen, eine Nachricht abgeschickt zu haben.

## ■ 2.2 Kryptographische Algorithmen

**Kryptographische Algorithmen** sind Berechnungsvorschriften, d. h. mathematische Funktionen zur Ver- und Entschlüsselung. Bei **symmetrischen Algorithmen** wird zum Chiffrieren und zum Dechiffrieren immer der gleiche Schlüssel  $K$  benutzt und es gilt

$$E_K(M) = C$$

$$D_K(C) = M$$

$$D_K(E_K(M)) = M.$$

Bei **asymmetrischen Algorithmen** wird zum Chiffrieren ein Schlüssel  $K_1$  und zum Dechiffrieren ein anderer Schlüssel  $K_2$  benutzt und es gilt:

$$E_{K_1}(M) = C$$

$$D_{K_2}(C) = M$$

$$D_{K_2}(E_{K_1}(M)) = M.$$

Man unterscheidet bei kryptographischen Algorithmen zwischen **Stromchiffren** und **Blockchiffren**. Bei Stromchiffren wird ein Zeichen nach dem anderen verschlüsselt. Bei Blockchiffren wird die Nachricht in Blöcke (z. B. der Länge 64 Bit) zerteilt und dann ein Block nach dem anderen verschlüsselt. Die Vereinigung von Algorithmus, zugehörigen Schlüsseln und den verschlüsselten Nachrichten wird **Kryptosystem** genannt.

Früher wurden so genannte **ingeschränkte Algorithmen** benutzt. Bei diesen hängt die Sicherheit davon ab, ob die Arbeitsweise des Algorithmus geheim ist. Die Geheimhaltung eines Algorithmus hat folgende schwerwiegenden Nachteile beim praktischen Einsatz:

- Verlässt eine Person eine Benutzergruppe (z. B. eine Firma), dann muss der Algorithmus geändert werden.
- Auch wenn der Quellcode der Programme nicht öffentlich bekannt ist, kann ein Angreifer aus den Maschinenprogrammen die Algorithmen rekonstruieren. Ingeschränkte Algorithmen können daher nicht an Dritte weitergegeben werden. Sie wären dann wertlos.
- Qualitätskontrolle von eingeschränkten Algorithmen findet in den meisten Fällen nicht in ausreichendem Maße statt, da die entwickelte Software nicht der Kritik und den Angriffen der Öffentlichkeit standhalten muss.

Heute werden Algorithmen mit **Schlüssel** benutzt. Der Schlüssel ist meist eine natürliche Zahl, dargestellt im Binärsystem, d. h. als Folge von Bits. Der Algorithmus ist idealerweise allgemein bekannt und nur der zugehörige Schlüssel muss geheim gehalten werden. Dieses Vorgehen wurde schon im 19. Jahrhundert von A. Kerkhoffs [Kah67] gefordert:

Die Sicherheit eines Verschlüsselungsverfahrens darf nur von der Geheimhaltung des Schlüssels abhängen, nicht jedoch von der Geheimhaltung des Algorithmus.

Kerkhoffs forderte damit, dass die Sicherheit eines Algorithmus nicht darunter leiden darf, dass er veröffentlicht wird. Die aktuelle Praxis in der Kryptographie zeigt deutlich, dass durch möglichst frühzeitige Offenlegung der Algorithmen die Sicherheit eines Kryptosystems erheblich größer wird. Denn sobald ein Algorithmus publiziert ist, muss er den Attacken der Experten standhalten, d. h. er muss sich bewähren. Sind über einen langen Zeitraum alle Attacken erfolglos, so stärkt dies das Vertrauen der Benutzer in die Sicherheit des Algorithmus. Diese Methodik der Entwicklung moderner Algorithmen ist ein wichtiger Bestandteil der so genannten **starken Kryptographie**.

In der Geschichte der Kryptographie gibt es viele Beispiele für die Verletzung von Kerkhoffs' Prinzip, was zu teilweise dramatischen Sicherheitslücken führte. Zwei Beispiele aus dem Jahr 1999 zeigen, dass selbst namhafte Firmen das Kerkhoffs-Prinzip nicht beachten. Im Online-Magazin der Zeitschrift c't vom 7.12.99<sup>2</sup> war folgender Text zu lesen:

#### **Handy-Verschlüsselung angeblich geknackt**

Die beiden israelischen Kryptologen Alex Biryukov und Adi Shamir haben Medienberichten zufolge den Verschlüsselungsalgorithmus geknackt, der GSM-Handy-Telefonate auf der Funkstrecke zur Mobiltelefon-Basisstation schützt. ...

Eines zeigen die Vorfälle um die GSM-Verschlüsselungsalgorithmen A5/1 und A5/2 aber schon jetzt deutlich: *Der Versuch, Krypto-Verfahren geheim zu halten, dient nicht der Sicherheit*. Das hat anscheinend auch die GSM-Association gelernt: Ihr Sicherheitsdirektor James Moran äusserte dem Online-Magazin Wired gegenüber, dass man künftige Algorithmen von vornherein offenlegen will, um der Fachwelt eine Prüfung zu ermöglichen. (nl/c't)

Eine Woche später, nämlich am 15.12.99<sup>3</sup> erschien an gleicher Stelle die nächste Meldung zu diesem Thema:

#### **Netscape verschlüsselt Passwörter unzureichend**

Der Netscape Navigator legt Passwörter für den Zugriff auf E-Mail-Server nur unzureichend verschlüsselt ab. Zwei Mitarbeiter des US-Softwarehauses Reliable Software Technologies (RST) brauchten lediglich acht Stunden, um den Algorithmus zu knacken. ...

Der Algorithmus zerhacke die Passwörter zwar, es handle sich jedoch um *keine starke Verschlüsselung*, so Gary McGraw von RST. Durch die Eingabe einfacher Passwörter wie „a“, „b“ und so weiter sei man relativ schnell dahinter gekommen.

...

Der US-Sicherheitsexperte Bruce Schneier wertet die Entdeckung als weiteres Beispiel dafür, *wie schädlich proprietäre Verschlüsselungsverfahren sein können*. (ad[2]/c't)

<sup>2</sup> Siehe <http://www.heise.de/newsticker/data/nl-07.12.99-000/>

<sup>3</sup> Siehe <http://www.heise.de/newsticker/data/ad-15.12.99-001/>

Ein weiteres aktuelles Beispiel betrifft das Verschlüsselungsprotokoll WEP (Wired Equivalent Privacy), das bei Funk-Netzwerken nach dem Standard IEEE802.11 verwendet wird. Die Autoren von [BGW01] schreiben

#### Conclusions

Wired Equivalent Privacy (WEP) isn't. The protocol's problems is a result of misunderstanding of some cryptographic primitives and therefore combining them in insecure ways. These attacks point to *the importance of inviting public review* from people with expertise in cryptographic protocol design; had this been done, the problems stated here would have surely been avoided.

Diese drei Meldungen sprechen für sich und bedürfen keines weiteren Kommentars.

## ■ 2.3 Kryptographische Protokolle

Ein kryptographischer Algorithmus zum Verschlüsseln kann auf vielfältige Art und Weise in unterschiedlichen Anwendungen eingesetzt werden. Damit eine Anwendung immer in der gleichen und korrekten Art abläuft, werden kryptographische Protokolle definiert.

Im Gegensatz zu den kryptographischen Algorithmen handelt es sich bei den Protokollen um Verfahren zur Steuerung des Ablaufs von Transaktionen für bestimmte Anwendungen, wie zum Beispiel das in Kapitel 1 vorgestellte Protokoll für elektronisches Bargeld.

## ■ 2.4 Public-Key-Algorithmen

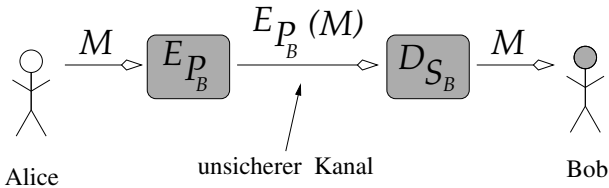
Wollen zwei Parteien über einen unsicheren Kanal mit einem symmetrischen Algorithmus geheime Nachrichten austauschen, so müssen sie einen geheimen Schlüssel vereinbaren. Wenn sie nur über einen unsicheren Kanal verfügen, sind sie mit dem Schlüsseltauschproblem (Kapitel 5) konfrontiert.

Erst Mitte der 70er Jahre wurde mit der Erfindung der Public-Key-Kryptographie eine befriedigende Lösung gefunden. Sie kam genau zum richtigen Zeitpunkt, um für eine sichere Kommunikation im Internet den Grundstein zu legen. Systeme wie zum Beispiel PGP [Zim95a] (Kapitel 8.1) zum Verschlüsseln von E-Mails wären undenkbar ohne Public-Key-Algorithmen.

Vor der Erfindung der Public-Key-Algorithmen beschränkte sich das Verschlüsseln von Nachrichten auf spezielle, zum Beispiel militärische Anwendungen, bei denen der hohe Aufwand für den Schlüsseltausch gerechtfertigt war. Mit Hilfe der Public-Key-Kryptographie kann nun jedermann mit beliebigen Partnern geheime Nachrichten austauschen, Dokumente signieren und viele andere kryptographische Anwendungen wie zum Beispiel elektronisches Bargeld nutzen.

Algorithmen mit öffentlichem Schlüssel sind asymmetrische Algorithmen, die einen geheimen Schlüssel  $S$  (secret key) sowie einen öffentlichen Schlüssel  $P$  (public key) benutzen, deren Arbeitsweise und Sicherheit in Kapitel 5 ausführlich untersucht wird.

Die Idee der Public-Key-Kryptographie ist in Bild 2.1 dargestellt. Wenn Bob<sup>4</sup> geheime Botschaften empfangen möchte, so erzeugt er einen öffentlichen Schlüssel  $P_B$ , den er all seinen Kommunikationspartnern zukommen lässt und einen geheimen Schlüssel  $S_B$ , den er sicher verwahrt.<sup>5</sup>



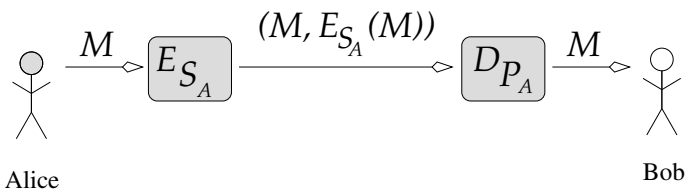
**BILD 2.1** Austausch einer Nachricht mit einem Public-Key-Verfahren. Es werden öffentlicher Schlüssel  $P_B$  und geheimer Schlüssel  $S_B$  von Bob benutzt

Will nun Alice eine geheime Nachricht an Bob schicken, so benutzt sie zum Verschlüsseln den öffentlichen Schlüssel  $P_B$  von Bob. Dieser dechiffriert die Nachricht dann mit seinem geheimen Schlüssel  $S_B$ . Zum Verschlüsseln wird nur der öffentliche Schlüssel benötigt. Mit ihm kann also jedermann eine verschlüsselte Nachricht an Bob schicken, aber nur Bob kann sie mit seinem geheimen Schlüssel lesen. Dieses Prinzip entspricht der Funktion vieler Wohnungstüren, bei denen das Schloss verriegelt, sobald die Türe geschlossen wird. Jedermann kann die Türe schließen. Das Öffnen von außen ist dagegen nur für den Besitzer des Schlüssels möglich.

Damit Bob auch tatsächlich den Original-Klartext liest, muss gelten:

$$\begin{aligned} E_{P_B}(M) &= C \\ D_{S_B}(C) &= M \\ D_{S_B}(E_{P_B}(M)) &= M. \end{aligned}$$

Beim Signieren eines Dokumentes  $M$  geht man umgekehrt vor wie beim Verschlüsseln. Im Prinzip verschlüsselt Alice das Dokument mit ihrem geheimen Schlüssel und hängt das Resultat als Signatur an das Dokument an. Wenn nun am Dokument oder an der Signatur auch nur ein Bit geändert wird, ist die Signatur ungültig (Kapitel 6).



**BILD 2.2** Alice signiert ein Dokument  $M$  mit ihrem geheimen Schlüssel  $S_A$  und Bob prüft die Signatur mit Alices öffentlichem Schlüssel  $P_A$

<sup>4</sup> „Alice“ und „Bob“ als Kommunikationspartner sind Bestandteil der kryptographischen Fachsprache.

<sup>5</sup> Zur Vermeidung von Missverständnissen sei hier schon bemerkt, dass der Empfänger eines öffentlichen Schlüssels  $P_B$  immer dessen Authentizität überprüfen muss (Kapitel 7).

# Index

## Symbole

- ⊕ 52, 190
- ⊗ 191
- $\varphi$ -Funktion, Eulersche 183

## A

- Additional-Decryption-Key 126
- ADK 126, 130
- Adleman, Leonard 79
- Advanced-Encryption-Standard 68
- AES 68, 99, 108, 113, 127, 136
- Affine Chiffre 35
- Algorithmus, randomisierter 195
- Alice 25
- Alphabet 21
- Anderson, Ross 75
- Angriff 26
- Angriff mit gekauftem Schlüssel 26
- Angriff mit Gewalt 26
- ANSI 60
- approximate entropy 197
- Arithmetik, modulare 175
- asymmetrischer Algorithmus 22
- AusweisApp-Software 135
- Authentifikation 22, 100
- Authentifikation, biometrische 114
- Authentifizierung 22
- Authentizität 91, 113

## B

- Babbage, Charles 40
- BBS-Generator 197
- Benutzerauthentifikation 100, 102, 112
- berechenbar 98
- Berechnungskomplexität 27
- Betriebsmodi von Blockchiffren 74
- Bigramme 36
- Biham, Eli 75
- binary symmetric source 195

- Biometrische Verfahren 114
- Bitcoin 151
- Bit-Commitment 141
- blenden 88
- Bletchley Park 48
- blinde Signatur 16, 107, 141
- Blockchiffre 22, 59
- Blowfish 75, 131
- Bob 25
- Bombe 49
- Brute-Force-Angriff 26, 28, 48
- BSI 110, 134
- BSS 195
- Bundesamt für Sicherheit in der Informationstechnik 110, 134

## C

- CA 119, 125, 133
- CBC 74
- CBC-Modus 74, 113
- Certification Authority 119, 125
- CESG 77
- Challenge-and-Response 102, 112, 132, 135
- Chaum, David 15
- Chiffretext 21
- Chiffriermaschine 31, 45
- Chinesischer Restsatz 81
- Chipkarte 88, 103, 109, 115, 173
- Chosen-Ciphertext-Angriff 26, 87
- Chosen-Plaintext-Angriff 26
- cipher block chaining 74
- ciphertext 21
- Ciphertext-Only-Angriff 26
- Clipper-Chip 167
- CMRK 126
- COCOM 169
- Codebuch 49
- Colossus 49
- Coppersmith, Don 75

Corporate-Message-Recovery-Key 126  
CRYPTO 103

**D**

Daemen, Joan 69  
Data-Encryption-Standard 59, 60  
Datenkomplexität 27  
decryption 21  
DES 59, 113, 127, 132  
DES40 132  
differential power analysis 88  
differentielle Kryptanalyse 67, 73  
Diffie, Whitfield 91  
Diffie-Hellman-Algorithmus 77, 91, 92, 94, 132, 136  
Diffusion 59  
Digicash 15  
Digital Signature Algorithm 106, 124  
Digital Signature Standard 107  
digitale Signatur 97  
Ding, Yan Zong 55  
diskreter Logarithmus 92, 93  
Divisionsrest 175  
DPA 88  
DSA 106, 124, 127, 131, 136  
DSS 107

**E**

eBay 149  
ECB 74  
ECB-Modus 74  
ECHELON 167  
echte Zufallszahl 196  
E-Commerce 145  
EES 167  
EFF 60  
Eingangsp permutation 61  
Einwegfunktion 97, 98  
Einweg-Hash-Funktion 74, 98, 99, 102, 132  
electronic codebook 74  
Electronic Commerce 15  
Electronic Frontier Foundation 60  
Electronic-Wallet 147  
elektronische Münze 15  
elektronische Signatur 111  
elektronische Signatur, fortgeschrittene 111  
elektronische Signatur, qualifizierte 112  
elektronische Unterschrift 97  
elektronisches Bargeld 24, 107, 139  
ElGamal 77, 106, 127  
ElGamal-Algorithmus 93  
Elliptische Kurven, Kryptographie mit 93, 136

Emacs 124  
E-Mail signieren 108  
encryption 21  
Enigma 45  
Entschlüsselung 21  
Escrowed Encryption Standard 167  
ETH 75  
Euklidischer Algorithmus, erweiterter 35, 183  
Eulersche  $\varphi$ -Funktion 183  
Eve 79

**F**

FAR 114  
Feistel, Horst 59  
Feistel-Chiffre 59, 75  
Fermatscher Satz 179  
Fingerabdruck, kryptographischer 98  
fingerprint 117, 119  
Firewall 125  
Fortezza 132  
Friedman, William 43  
Friedman-Test 40, 43  
FRR 114  
ftp 132

**G**

Galoiskörper 94  
Galoistheorie 98  
Geburtstagsangriff 100, 216  
Geheimdienste 167  
geheimer Schlüssel 24, 108  
Geheimtext 21  
Geldkarte 146  
GF( $2^8$ ) 70, 75, 190  
GF( $2^n$ ) 94  
globale Deduktion 26  
GNU Privacy Assistant 128  
GNU Privacy Projekt 127  
GnuPG 87, 127  
GnuPP 127  
Gnu-Privacy-Guard 127  
GPA 128  
Gruppe 94

**H**

Hammingabstand 66  
Hash-Wert 98, 132  
Hellman, Martin 91  
Herausfordern und Antworten 102  
Homophone Chiffre 37  
HTTP-Verbindungen 132  
hybride Verschlüsselung 123

**I**

IDEA 75, 131, 132  
IDS 125  
IETF 131  
Inflation 147  
Informationsdeduktion 26  
Initialisierungsvektor 75  
injektiv 21  
Injektivität 21  
Integrität 22, 113  
Internet Engineering Task Force 131  
Internet Mail Consortium 131  
Internet-Pakete 133  
intrusion detection system 125  
IP 133  
IP Security 133  
IPSec 133  
IP-Tunneling 132  
ISO-Schichtenmodell 132  
ITAR 169

**K**

Kasiski, Friedrich 40  
Kasiski-Test 40  
Kerchoffs-Prinzip 23, 48, 147, 195  
key-recovery 167  
Key-Server 117  
KISS 197  
Klartext 21  
klassische Chiffren 31  
Klíma, Vlastimil 129  
Known-Plaintext-Angriff 26, 49, 68  
Knudsen, Lars 75  
Knuth, Don 195  
Kocher, Paul 87  
Koinzidenzindex 43  
Kolmogorov-Komplexität 196, 201  
Konfusion 59  
Kongruenz 175  
Kryptanalyse 21, 40  
Kryptographie 21  
kryptographischer Algorithmus 15, 22  
kryptographisches Protokoll 15, 24  
Kryptologie 21  
Kryptosystem 22  
kubische Gleichung 103

**L**

Lai, Xuejia 75  
Lawineneffekt 66  
LDAP 121  
LFSR 198

Lightweight Directory Access Protocol 121  
linear feedback shift register 198  
lineare Komplexität 201  
lineare Kryptanalyse 73  
linearer Kongruenzgenerator 197  
lineares Schieberegister mit Rückkopplung 198  
lokale Deduktion 26  
LUCIFER 59

**M**

MAC 74, 113, 168  
Macro-Payment 145  
mailcrypt 124  
Mallory 90  
Man-in-the-Middle 90, 117  
Man-in-the-Middle-Angriff 90  
MARS 69, 75  
Massey, James 75  
Mauborgne, J. 52  
Maurer, Ueli 56  
MD4 99  
MD5 99, 127, 132  
Meet-in-the-Middle-Angriff 68, 76  
message digest 99  
Message-Authentication-Code 74, 113  
Micro-Payment 15, 139, 145  
Miller 188  
modulare Arithmetik 31, 175  
modulo 175  
Mondex-System 147  
monoalphabetisch 31  
monoalphabetische Chiffre 31, 36  
MPI-Format 130  
Multimedia-Dokumente 97  
Multiplikative Chiffre 33  
multi-precision integer 130

**N**

Nachricht 21  
National Institute of Standards and Technology 60  
National Security Agency 60  
Neumann, John von 201  
Neumann-Filter 201  
NFS 82  
NIST 60, 99, 106, 167  
nPA → Personalausweis, neuer  
NSA 60, 77, 99, 106, 167  
number field sieve 82

**O**

On-Card Matching 115  
One-Time-Pad 52, 58, 73, 198, 200



Online-Banking 97  
 OpenPGP-Standard 127, 129, 131

## P

padding 75  
 passphrase 108, 132  
 Passwort 108  
 Passwortverschlüsselung 100  
 PayPal 149  
 Peggy 103  
 Periodendauer 199  
 Personalausweis, neuer 109, 121, 134  
 Personalisierung 119  
 PGP 87, 107, 111, 117, 120, 123  
 PIN-Code 109, 115, 135  
 PKCS 130  
 PKI 117, 134  
 plaintext 21  
 Point-of-Sale-Händlerterminal 150  
 polyalphabetisch 31  
 Polyalphabetische Chiffre 37  
 POSH 150  
 pretty good privacy 123  
 PRNG 195  
 Probabilistische Verschlüsselung 87  
 Protokoll, kryptographisches 24  
 Prüfsumme 113  
 pseudo random number generator 195  
 Pseudozufallszahlengenerator 55, 74, 195, 197  
 public key 24  
 public key cryptography system 130  
 Public-Key-Infrastruktur 91, 117, 130, 134  
 Public-Key-Kryptographie 24, 77, 97  
 Public-Key-System 120

## Q

Quantenkryptographie 91  
 Quantenschlüsseltausch 91

## R

Rabin, Michael 55, 188  
 RC2 75, 132  
 RC4 132  
 RC5 75  
 RC6 69, 75  
 real random number generator 196  
 Reflektor 45  
 relativ prim 183  
 Replay-Angriff 102, 113  
 Rest 175  
 RFID Chip 134  
 Rijmen, Vincent 69

Rijndael 69, 94  
 RIPE-MD 99  
 RIPEMD160 127  
 Ritter, Terry 7  
 Rivest, Ron 79, 167  
 ROC-Kurve 114  
 root-CA 120  
 Rosa, Tomáš 129  
 RRNG 196  
 RSA 106, 127, 131, 132  
 RSA-Algorithmus 77, 79, 94, 108, 186  
 RSA-Algorithmus, Korrektheit 81  
 RSA-Algorithmus, Sicherheit 82  
 Rucksack-Algorithmus 77  
 Runden 61

## S

S/MIME 117, 130  
 S-Box 75  
 S-Box-Transformation 64  
 Scherbius, Arthur 45  
 Schieberegister 198  
 Schlüssel 21, 23  
 Schlüssel, geheimer 24, 108  
 Schlüssel, öffentlicher 24  
 Schlüssel, schwacher 64  
 Schlüsselraum 29  
 Schlüsselring 126  
 Schlüsseltausch 89  
 Schlüsseltauschproblem 24, 77  
 Schlüsselwort 38  
 Schlüsselwortlänge 40, 43  
 Schlusspermutation 61  
 Schneier, Bruce 60, 75  
 schwacher Schlüssel 64  
 Schwellenwertproblem,  $(m, n)$  140, 143  
 secret key 24  
 Secret-Splitting 139, 143  
 Secret-Splitting-Protokoll 141  
 Secure Electronic Transactions 148  
 secure shell 131  
 Secure socket layer 132  
 Secure-Hash-Algorithm 99  
 Secure-Hash-Standard 102  
 seed 195, 197  
 Seed-Zahl 197  
 Seiteneffekt 173  
 Seiteneffekt, Angriff 87  
 selbstinverse Abbildung 47  
 Serpent 69, 75  
 SET 145, 148  
 SHA 99, 102

SHA-1 99, 127, 132  
Shamir, Adi 79  
Shannon, Claude 59  
SHS 102  
sicherer Algorithmus 27  
Sicherheitsschlüssel 27  
Sieb des Eratosthenes 82  
Signatur, blinde 107  
Signatur, digitale 97  
Signaturgesetz 97, 109, 135  
simple power analysis 88  
Smart Contract 151  
Speicherplatzbedarf 27  
Spionage 49  
Spruchschlüssel 50  
SSH 131, 137  
SSL 120, 132, 150  
starke Kryptographie 23, 115  
statistische Analyse 32  
Steckbrett 45  
Steganographie 21, 168  
Stromchiffre 22, 52, 74, 200  
Substitutionschiffre 31  
symmetrischer Algorithmus 22

**T**

Tartaglia, Niccolò 103  
Tauschchiffre 35  
TCP/IP-Port 132  
teilerfremd 183  
thermisches Rauschen 54  
TIGER/192 127  
timing-attack 87  
Transport Layer Security 132  
Transpositionschiffre 31  
Trent 89  
Triple-DES 61, 68, 131, 132  
Trojanerangriff 115, 173  
Trojanisches Pferd (Angriff) 115, 173  
Trustcenter 89, 117, 118, 125  
Trustcenter, akkreditiertes 112  
Tunneln 132  
Turing 52  
Turing, Alan 48  
TWOFISH 127  
Twofish 69, 75

**U**

U-Boot 48  
uneingeschränkt sicherer Algorithmus 27  
Unterschrift, elektronische 97

**V**

Verbindlichkeit 22  
verborgene Parameter 196  
Vernam, G. 52  
Verschiebechiffre 31, 32  
Verschlüsselung 21  
Victor 103  
Vigenère, Blaise de 38  
Vigenère-Chiffre 38  
Viren 120  
virtual private networking 125, 133  
Vollständiges Aufbrechen 26  
von-Neumann-Rechner 54  
VPN 125, 133

**W**

wahrscheinliche Wörter, Methode 49  
Walze 45  
Walzenlage 50  
Wassenaar Abkommen 169  
Web-Browser 132  
Web-of-Trust 120, 126  
white card 146  
Wörterbuchangriff 100

**X**

X.509 130

**Z**

Zahlentheorie 175  
Zahlkörpersieb 82, 89  
Zeitstempel 119  
zentralen Kreditausschuss 146  
Zero-Knowledge-Beweis 103, 104, 113  
Zero-Knowledge-Protokoll 102, 103  
Zertifikat 118, 130  
Zertifikathierarchie 119  
Zertifizierung 118  
Zimmermann, Phil 123  
ZKA 146  
zufällig 195  
Zufallsbitfolge, echte 53  
Zufallszahl 195  
Zustand 70  
Zyklus 50