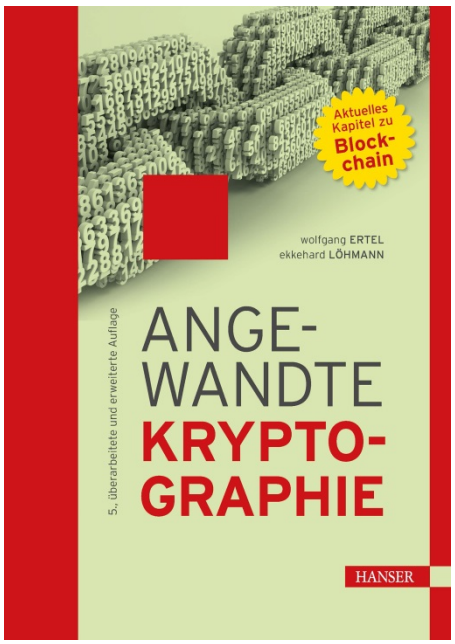
HANSER



Leseprobe

zu

Angewandte Kryptographie

von Wolfgang Ertel und Ekkehard Löhmann

5., überarbeitete und erweiterte Auflage
Mit 53 Bildern, 55 Aufgaben und 21 Tabellen

ISBN (Buch): 978-3-446-45468-2
ISBN (E-Book): 978-3-446-45704-1

Weitere Informationen und Bestellungen unter
http://www.hanser-fachbuch.de/978-3-446-45468-2
sowie im Buchhandel

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.

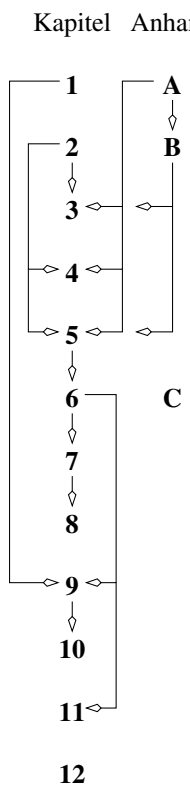


BILD 1 Kapitelstruktur

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

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, 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

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 Löhmann, 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

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** vereinigt Kryptographie und Kryptanalyse.

Bei der **Steganographie** werden geheime Nachrichten nicht verschlüsselt, sondern versteckt. Historisches Beispiel hierfür sind unsichtbare Geheimtinten, die später durch Erwärmen sichtbargemacht 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[1]. 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.

[1] Eine Abbildung $f: D \to B$ 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

$$\begin{aligned} E_K(M) &= C \\ D_K(C) &= M \\ D_K\left(E_K(M)\right) &= M\,. \end{aligned}$$

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

$$\begin{aligned} E_{K_1}(M) &= C \\ D_{K_2}(C) &= M \\ D_{K_2}\left(E_{K_1}(M)\right) &= M\,. \end{aligned}$$

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 **eingeschränkte 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. Eingeschränkte 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[2] 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 vorneherein offenlegen will, um der Fachwelt eine Prüfung zu ermöglichen. (nl/c't)

Eine Woche später, nämlich am 15.12.99[3] 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)

[2] Siehe http://www.heise.de/newsticker/data/nl-07.12.99-000/
[3] 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[4] 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.[5]

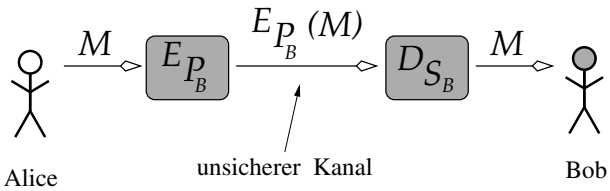


BILD 2.1 Austausch einer Nachricht mit einem Public-Key-Verfahren. Es werden öffentlicher Schlüssel P_B und geheimer Schlüssel S_B von Bob benutzt.

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:

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

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).

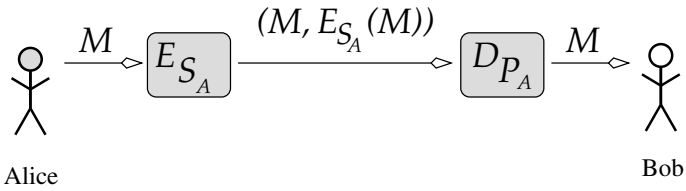


BILD 2.2 Alice signiert ein Dokument M mit ihrem geheimen Schlüssel S_A und Bob prüft die Signatur mit Alices öffentlichem Schlüssel P_A.

[4] „Alice“ und „Bob“ als Kommunikationspartner sind Bestandteil der kryptographischen Fachsprache.

[5] Zur Vermeidung von Missverständnissen sei hier schon bemerkt, dass der Empfänger eines öffentlichen Schlüssels P_B immer dessen Authentizität überprüfen muss (Kapitel 7).

Für die Sicherheit von Signatur und Verschlüsselung ist es sehr wichtig, dass es praktisch unmöglich ist, aus einem öffentlichen Schlüssel P_A den zugehörigen geheimen Schlüssel S_A zu berechnen (Kapitel 5).

2.5 Kryptanalyse

Das Aufbrechen oder Knacken eines Kryptosystems lässt sich in verschiedene Kategorien einteilen. Beim **vollständigen Aufbrechen** wird der Schlüssel K gefunden und mit $D_K(C) = M$ kann jede Nachricht entschlüsselt werden. Das Finden eines zu D_K äquivalenten alternativen Algorithmus ohne Kenntnis des Schlüssels wird **globale Deduktion** genannt. Das Finden des Klartextes für nur einen abgefangenen Chiffretext wird **lokale Deduktion** genannt. Noch schwächer ist die **Informationsdeduktion**, bei der nur eingeschränkte Informationen über den Schlüssel oder den Klartext ermittelt werden können.

Es gibt verschiedene Arten von kryptanalytischen **Angriffen** auf ein Kryptosystem, von denen die wichtigsten hier erwähnt seien:

Ciphertext-Only-Angriff: Der Kryptanalytiker verfügt nur über eine bestimmte Menge Chiffretext.

Known-Plaintext-Angriff: Der Kryptanalytiker kennt zusätzlich den zum Chiffretext gehörenden Klartext.

Chosen-Plaintext-Angriff: Der Kryptanalytiker kann einen beliebigen Klartext vorgeben und hat eine Möglichkeit zu diesem vorgegebenen Text an den Chiffretext zu gelangen. Ein Chosen-Plaintext-Angriff ist beispielsweise möglich zum Knacken von Public-Key-Systemen, denn mit dem öffentlichen Schlüssel kann ein Angreifer jeden beliebigen Klartext verschlüsseln.

Chosen-Ciphertext-Angriff: Der Kryptanalytiker kann einen beliebigen Chiffretext vorgeben und hat eine Möglichkeit, an den zugehörigen Klartext zu gelangen.

Angriff mit Gewalt: Der Kryptanalytiker bedroht oder foltert eine Person mit Zugang zum Schlüssel.

Angriff mit gekauftem Schlüssel: Der Schlüssel wird mittels Bestechung „gekauft“.

Die beiden letztgenannten Angriffe sind sehr wirkungsvoll und gerade bei den Verfahren der starken Kryptographie für den Angreifer oft der einzige Weg zum Schlüssel. Aus diesem Grund gilt:

> Bei den Verfahren der starken Kryptographie stellt meist der Mensch als Besitzer des Schlüssels die größte Sicherheitslücke dar.

Ein Angriff, bei dem alle möglichen Schlüssel ausprobiert werden, wird **Brute-Force-Angriff** (Angriff mit Brachialgewalt) genannt. Bei einem Ciphertext-Only-Angriff probiert man (normalerweise der Computer) so lange, bis der berechnete Klartext Sinn macht. Dies zu überprüfen kann unter Umständen nicht ganz einfach sein. Bei einem Known-Plaintext-Angriff verschlüsselt man so lange die Klartexte bis der berechnete Chiffretext mit

dem bekannten Chiffretext übereinstimmt. Mit einem Brute-Force-Angriff lassen sich – bei entsprechend hohem Aufwand – fast alle Kryptosysteme knacken. Trotzdem besteht kein Grund zur Besorgnis, denn der für einen erfolgreichen Brute-Force-Angriff nötige Aufwand lässt sich durch Wahl eines langen Schlüssels beliebig hoch treiben (siehe Abschnitt 2.6). Für die Beurteilung der benötigten Schlüssellänge ist folgende Definition sehr hilfreich.

Definition 2.1

Ein Algorithmus gilt als **sicher**, wenn

- der zum Aufbrechen nötige Geldaufwand den Wert der verschlüsselten Daten übersteigt oder
- die zum Knacken erforderliche Zeit größer ist als die Zeit, die die Daten geheim bleiben müssen, oder
- das mit einem bestimmten Schlüssel chiffrierte Datenvolumen kleiner ist als die zum Knacken erforderliche Datenmenge.

Ein Algorithmus ist **uneingeschränkt sicher**, wenn der Klartext auch dann nicht ermittelt werden kann, wenn Chiffretext in beliebigem Umfang vorhanden ist.

In Abschnitt 3.8 werden wir einen uneingeschränkt sicheren Algorithmus kennen lernen.

Zur Beurteilung des Aufwands für einen Angriff sollten drei Größen berechnet werden. Das in den meisten Fällen wichtigste Maß ist die **Berechnungskomplexität**, das heißt die Rechenzeit in Abhängigkeit von der Schlüssellänge. Weitere Kriterien sind der **Speicherplatzbedarf** sowie die **Datenkomplexität**, welche die Menge der benötigten Eingabedaten (z. B. des abgehörten Chiffretextes) angibt.

2.6 Sicherheit von Schlüsseln

Die Sicherheit aller kryptographischen Verfahren basiert im Wesentlichen auf der Schwierigkeit, einen geheimen Schlüssel zu erraten oder ihn auf anderem Wege zu beschaffen. Es ist durchaus möglich, einen Schlüssel zu erraten, wenn auch die Wahrscheinlichkeit mit wachsender Schlüssellänge sehr klein wird.

Absolute Sicherheit gibt es in der Kryptographie nicht. Jedoch gibt es in unserer Welt an keiner Stelle absolute Sicherheit. Auch das in Fort Knox deponierte Gold kann mit entsprechendem Aufwand gestohlen werden. Trotzdem ist es sicher genug verwahrt. Dass auch die in der Kryptographie benutzten Schlüssel „sicher genug“ sind, sollen die folgenden Beispiele zeigen. Der Einfachheit halber beschränken wir uns hier auf Brute-Force-Angriffe.[6]

Zuerst beurteilen wir die Sicherheit der beiden in Bild 2.3 dargestellten Schlüssel. Der linke, ein so genannter Schubschlüssel, hat an sechs Positionen eine Ausfräsung oder keine. Das ergibt $2^6 = 64$ verschiedene Schließungen. Der Sicherheitsschlüssel rechts hat sechs

[6] Für die meisten Verschlüsselungsverfahren gibt es effektivere Angriffe, auf die wir später noch eingehen werden. Trotzdem bedeutet ein längerer Schlüssel mehr Sicherheit.

BILD 2.3 Einfacher Schubschlüssel und Sicherheitsschlüssel

Einkerbungen mit fünf möglichen Tiefen und zusätzlich noch sechs binäre Bohrungen, woraus sich $5^6 2^6 = 10^6 = 1\,000\,000$ verschiedene Schließungen ergeben. Für einen Brute-Force-Angriff mit sturem Probieren aller Schlüssel bis zum Erfolg würde ein Einbrecher bei dem Schubschloss etwa eine Minute benötigen, wenn man pro Schlüssel zwei Sekunden ansetzt, wobei er im Mittel nur etwa die Hälfte aller Schlüssel testen muss. Bei dem Sicherheitsschloss wäre der Einbrecher im Mittel etwa 1 000 000 Sekunden, d. h. zwölf Tage beschäftigt, ganz zu schweigen von dem Aufwand für die Herstellung der Schlüssel und dem Transport von etwa 20 Tonnen Schlüssel. Sicherheitsschlösser kann man also als sicher bezüglich einem Brute-Force-Angriff bezeichnen. Man sollte jedoch bedenken, dass ein erfahrener Experte bei bestimmten Schließsystemen, allein durch das genaue Ansehen eines Schlüssels, dessen Code ablesen kann und damit in der Lage ist, den Schlüssel zu kopieren.

Starten wir nun mit einem digitalen 56 Bit langen Schlüssel, wie er zum Beispiel von DES (Abschnitt 4.1) benützt wird. Der Schlüsselraum hat die Größe $2^{56} \approx 7 \cdot 10^{16}$. Die beste bekannte Hardware-Implementierung von DES kann etwa $9 \cdot 10^{10}$ Schlüssel pro Sekunde testen, woraus sich eine mittlere Zeit von etwa $3.9 \cdot 10^5$ Sekunden ≈ 4.5 Tage ergibt. Bezüglich eines derartigen Brute-Force-Angriffs ist DES also nicht mehr sicher. Erhöhen wir die Schlüssellänge auf 100 Bit, so erhalten wir $2^{100} \approx 1.3 \cdot 10^{30}$ Schlüssel und die Zeit zum Knacken bei gleichen Annahmen liegt nun bei etwa $7 \cdot 10^{18}$ Sekunden $\approx 2 \cdot 10^{11}$ Jahre. Dies ist zwanzigmal länger als das Alter des Universums mit etwa 10^{10} Jahren.

Noch viel länger würde das sture Probieren bei einem Schlüssel der Länge 1024 Bit wie in Bild 2.4 dauern. Das Berechnen der genauen Zeit sei dem Leser als Übung überlassen.

```
0100001011011110101000111011000111111101010011001111001000001101111 01
0011111100101000001101100010001101000101001000110101101110110110100 10
0000011000100001001011101111000011111010100001111000011011010001110000
0101010011101011011111101110111111011011011001100011110001011110011100
0011111100010000001100010110101100101101111101100111000111001100110100
1111011010110101010010101000100001010111001001000110000001100101011 01
0001101001111110001111010001000011100101010000101000010100100001111 11
0110010100001000101000010111001000000011000100111101000101011100010 01
0010101101111100010111110101001010111000010001001101011001001111011 11
0001000101111010000100000100010011011011111000111011100001000101001 11
1000011011100111110001100111100101110111011011011100011101010011110 11
1100101011011100011001011011010111000010010101011100010111010100100 01
0000011010011011101000001101000000000011111100110110001001101010000 01
0011111101001100010101010011111111101011000111111001011001110111101 10
1101111110101001100011110011000101010000010000000100101011
```

BILD 2.4 Einer der 2^{1024} Schlüssel mit 1024 Bit

Hier erkennt man deutlich, dass entsprechend Definition 2.1 ein digitaler Schlüssel jede erdenkliche Sicherheitsstufe erreichen kann, wenn man ihn nur lang genug macht. Mit anderen Worten:

Der **Schlüsselraum**, d. h. die Menge, aus der ein Schlüssel gewählt wird, sollte möglichst groß sein. Er sollte mindestens so groß sein, dass der Aufwand für einen Angriff unakzeptabel hoch wird. ■

Obwohl ein Schlüssel mit 100 Bit ausreichende Sicherheit gegen einen Brute-Force-Angriff bietet, werden bei den Public-Key-Algorithmen heute tatsächlich Schlüssel mit 1024 Bit eingesetzt, denn es gibt effektive Angriffe, die eine derart große Schlüssellänge erfordern (Kapitel 5). Ähnliches gilt für fast alle Algorithmen. Daher ist es wichtig, ein möglichst gutes mathematisches Modell für die benutzten Algorithmen zu haben, um den Aufwand für diverse Angriffe abschätzen zu können.

Übungen

Aufgabe 2.1

Berechnen Sie die mittlere Zeit für das Knacken des 1024-Bit-Schlüssels einer 1024-Bit-Blockchiffre mit einem Brute-Force-Angriff unter der Annahme, dass Sie einen Block von 1024 Bit im Klartext und im Chiffretext vorliegen haben. Nehmen Sie an, Sie haben Zugriff auf einen Rechner, der pro Sekunde 1 Megabit verschlüsseln kann.

Aufgabe 2.2

Überlegen Sie sich ein Beispiel für eine nicht injektive Funktion zum Verschlüsseln eines Textes. Welches Problem ergibt sich?

Aufgabe 2.3

Gegeben sei folgende Chiffre C (ohne Schlüssel!), welche vom Alphabet $A = \{a,b,c\}$ auf das Alphabet $B = \{u,v,w,x,y,z\}$ abbildet. Die Vorschrift lautet

$$a \mapsto u \vee x\,, \qquad b \mapsto v \vee y\,, \qquad c \mapsto w \vee z\,,$$

wobei $a \mapsto u \vee x$ bedeutet, dass a zufällig entweder auf u oder x abgebildet wird. Es handelt sich hier übrigens um eine polyalphabetische Chiffre (siehe Definition 3.1).

a) Wie viele verschiedene Chiffretexte gibt es für *aabba*? Wie viele Klartexte sowie Chiffretexte der Länge k gibt es insgesamt?

b) Ist diese Abbildung umkehrbar? Wenn ja, geben Sie bitte die Dechiffrierfunktion an. ■

Index