

Kryptographie-Grundlagen

› Die Kryptographie hat längst die Grauzone des Spionagebereichs überschritten und soll für sichere Transaktionen im Internet sorgen. Als Grundlage dienen verschiedenste Verschlüsselungsverfahren.

› VON KLAUS MANHART und ULRICH PROELLER

Verschlüsselungsverfahren kommt im Rahmen der Datenübertragung eine besondere Bedeutung zu. Die Kryptographie soll die Geheimhaltung von Daten ermöglichen. Schließlich hat jede Person und jede Organisation ein legitimes Interesse an dem Schutz seiner Daten vor Ausspähung, sei es im Bereich von vertraulichen Bank- und Börsengeschäften oder sei es die E-Mail mit der Einladung zu einem Bewerbungsgespräch, die der bisherige Arbeitgeber nicht zu Gesicht bekommen soll. Insbesondere Firmen sind darauf angewiesen, ihre Einkaufskonditionen oder ihre Forschungsergebnisse vor den Augen der Konkurrenz zu schützen.

Neben dem offensichtlichen Zweck der Geheimhaltung muss die Kryptographie andere, grundlegende Kriterien erfüllen. Die Authentifizierung, die Integrität und die Verbindlichkeit beim Austausch von empfindlichen Daten sind vor allem für Geschäftsabschlüsse im Internet zwingend erforderlich.

› (Un-)Sicherheitsfaktoren

Die Authentifizierung spielt bei Internettransaktionen eine gewichtige Rolle: Erst der sichere Beweis, dass eine Person auch wirklich die ist, die sie zu sein vorgibt, führt Kunde und Verkäufer zu befriedigenden Geschäftsabschlüssen. Ein kritisches Gebiet findet sich außerdem im Bankgewerbe: Bank und Kunde müssen darauf vertrauen können, dass nur der Kontoinhaber Auskunft über seinen Kontostand bekommt und sich kein Unbefugter unter falschem Namen anmelden kann.

Ein weiterer Unsicherheitsfaktor ist die Integrität der ausgetauschten Daten: Bei abgeschlossenen Transaktionen muss der Empfänger einer Nachricht davon ausgehen können, dass die Nachricht auf dem Weg zu ihm nicht manipuliert wurde. Es wäre fatal, wenn sich bei einer elektronischen Überweisung der Empfänger des Geldes nachträglich verändern ließe.

Erst die Verbindlichkeit sichert einen gelungenen Geschäftsabschluss. Der Absender einer Nachricht darf später nicht leugnen können, dass die Nachricht, zum Beispiel eine Bestellung, tatsächlich von ihm stammt. Damit das Internet als Umschlagplatz für Waren und Dienstleistungen in großem Umfang genutzt werden kann, braucht es deshalb die verbindliche, [elektronische Unterschrift](http://www.tecchannel.de/internet/402/index.html) (<http://www.tecchannel.de/internet/402/index.html>). Sie wird in Zukunft der wohl wichtigste Anwendungsfall für starke Kryptographie sein.

› Starke Kryptographie

Vornweg gilt es, ein mögliches Missverständnis zu klären. Kryptographie, wie sie hier verstanden wird, hat nichts mit dem Verstecken von Daten ("Security by obscurity") zu tun, wie es zum Beispiel im Bereich der [Steganographie](http://www.tecchannel.de/multimedia/377/index.html) (<http://www.tecchannel.de/multimedia/377/index.html>) angewandt wird.

Die berechnungssichere, so genannte *starke Kryptographie* zeichnet sich im Wesentlichen dadurch aus, dass ihre Algorithmen publiziert und allgemein bekannt sind. Die Entschlüsselung der verschlüsselten Nachricht ist dabei in vertretbarer Zeit ohne Kenntnis des Schlüssels nicht möglich. Die Publikation der Ver- und Entschlüsselungsalgorithmen ermöglicht es den Kryptoanalytikern in aller Welt, das Verfahren auf Herz und Nieren zu überprüfen. Nur ein Algorithmus, der seit einigen Jahren publiziert ist und untersucht wurde, kann als sicher gelten, sofern keine

Schwachstellen gefunden wurden.

Grundsätzlich sind alle gängigen Kryptoalgorithmen durch Ausprobieren zu überwinden. Ob ein Kryptoalgorithmus sicher ist, hängt in der Praxis davon ab, ob der zum Knacken des Algorithmus notwendige Aufwand in Relation gesehen höher ist als der Wert der verschlüsselten Nachricht. Wenn das Ausprobieren selbst mit den schnellsten Computer weitaus länger dauert als die zu lesende Nachricht bedeutsam ist, kann von einem sicheren Algorithmus gesprochen werden. So ist zum Beispiel die Geheimhaltung der Konstruktionspläne eines neuen Autos spätestens nach dessen Markteinführung bedeutungslos. Ein Kryptoalgorithmus, bei dem das Entschlüsseln durch Ausprobieren mehr als zehn Jahre dauert, wäre in diesem Falle also sicher.

› Informationstheorie

Die hohe Redundanz menschlicher Sprache ist eine wichtige Voraussetzung für problemlose Verständigung, weshalb wir unseren Gesprächspartner auch verstehen, wenn es um uns herum sehr laut ist und die Hälfte des Satzes im Lärm untergeht. Wird eine Nachricht per Computer übertragen, ist diese hohe Redundanz unnötig.

Den Unterschied macht folgendes Beispiel deutlich, das den Informationsgehalt eines Satzes hinterfragt. Der Satz: "Ich lese gerade diesen tecChannel-Artikel" besteht (einschließlich Leerzeichen) aus 41 Buchstaben. In der üblichen ASCII-Kodierung würde er 41 Byte, das entspricht 328 Bit, belegen. Tatsächlich beträgt der Informationsgehalt eines Buchstabens gewöhnlicher Sprache statt 8 Bit aber nur 1,0 bis 1,5 Bit, da nicht alle Buchstaben des ASCII-Zeichensatzes vorkommen beziehungsweise gleich häufig sind. Der Informationsgehalt dieses Satzes liegt somit bei etwa 60 Bit. Der Rest, zirka 270 Bit, ist Redundanz, also überflüssige Information.

Die Redundanz macht die zu übertragende Datenmenge nicht nur größer als sie sein müsste. Sie bietet vor allem Kryptoanalytikern einen hervorragenden Ansatz für das Brechen der Verschlüsselung. Denn in jeder Sprache kommen verschiedene Buchstaben unterschiedlich häufig vor. Vor allem bei langen Chiffretexten ist es daher oft möglich, durch ausgefeilte statistische Analysen die Verschlüsselung zu knacken. Um dies unmöglich zu machen, komprimieren moderne Verschlüsselungsverfahren den Text, bevor sie ihn chiffrieren. Die Kompression entfernt einen großen Teil der Redundanz des Textes und macht so *statistische Kryptoanalyseverfahren* weitgehend sinnlos.

› Knackpunkt Rechenpower

Kryptoalgorithmen, bei denen der Aufwand zum Knacken der Verschlüsselung exponentiell mit der Schlüssellänge ansteigt, bieten einen ausreichenden Schutz vor dem wissenschaftlichen und technischen Fortschritt. Denn dieser ist, so paradox dies auf den ersten Blick klingen mag, der größte Feind der Kryptographie. Alle Aussagen über die Sicherheit von kryptographischen Verfahren beruhen auf Abschätzungen zum Rechenaufwand, der erforderlich ist, die Verschlüsselung zu brechen.

Diese Abschätzungen basieren auf der Geschwindigkeit heutiger Rechner und den bekannten mathematischen Verfahren. Die Entwicklung der Verarbeitungsgeschwindigkeit neuer Prozessoren und Rechner lässt sich noch halbwegs vorhersagen. Hier ist mit einer Verzehnfachung der Rechenleistung alle fünf Jahre zu rechnen. Dies gilt aber nur für die heute bekannten, siliziumbasierten Computer. Optische oder biologische Rechner der Zukunft ermöglichen durch massive Parallelverarbeitung eventuell um Zehnerpotenzen höhere Rechengeschwindigkeiten.

Ein ebenso großer Unsicherheitsfaktor ist die künftige Entwicklung der Mathematik. So glaubte man lange Zeit, dass das quadratische Sieb (QS) asymptotisch genauso schnell ist wie jede andere Faktorisierungsmethode. Mit NFS (Number Field Sieve) wurde eine Faktorisierungsmethode entdeckt, die potenziell bis zu zehn Mal schneller ist als das quadratische Sieb.

› Schlüssellängen

Die Frage nach der richtigen Schlüssellänge lässt sich nicht allgemein beantworten. Es kommt darauf an, wie wertvoll die Daten sind und wie lange sie geheim bleiben müssen.

Eine Sensationsmeldung im Journalismus steht am nächsten Tag in der Zeitung, sie muss also nur bis zur Auslieferung der Zeitung geschützt werden. Dagegen soll die Identität eines Spions auch nach 50 Jahren geheim bleiben. Eine kleine Zusammenstellung minimaler symmetrischer Schlüssellängen findet sich bei [Schneier](http://192.168.10.229/internet/416/18.html) (<http://192.168.10.229/internet/416/18.html>) :

Empfohlene Schlüssellängen

Informationsart	Lebensdauer	Minimale symmetrische Schlüssellänge
Militärtaktische Informationen	Minuten/Stunden	56-64 Bit
Produktankündigungen, Firmenzusammenschlüsse, Zinssätze	Tage/Wochen	64 Bit
Langfristige Geschäftsplanungen	Jahre	64 Bit
Wirtschaftsgeheimnisse (z.B. Coca-Cola-Rezept)	Jahrzehnte	112 Bit
Geheime Daten zur Wasserstoffbombe	Über 40 Jahre	128 Bit
Personenbezogene Daten	Über 50 Jahre	128 Bit
Geheimdiplomatie	Über 65 Jahre	Mindestens 128 Bit
Daten der US-Volkszählung	100 Jahre	Mindestens 128 Bit

Die angegebenen Schlüssellängen gelten für Schlüssel zu symmetrischen Verfahren. Die für Public-Key-Verfahren verwendeten Schlüssel müssen deutlich länger sein, um die gleiche Sicherheit zu gewährleisten.

Sicherheitsgewährleistung

Symmetrische Schlüssellänge	Asymmetrische Schlüssellänge
56 Bit	384 Bit
64 Bit	512 Bit
80 Bit	768 Bit
112 Bit	1792 Bit
128 Bit	2304 Bit

Längere Schlüssel erhöhen zwar die für das Ver- beziehungsweise Entschlüsseln benötigte Rechenzeit, doch diese Zeiten sind in der Regel so kurz, dass sie nicht ins Gewicht fallen. Es spricht daher wenig dagegen, lange bis sehr lange Schlüssel zu wählen. Es ist niemals sicher auszuschließen, dass die mathematische Wissenschaft oder die Entwicklung neuer, hochspezialisierter Chips zur Kryptoanalyse vermeintlich sichere Schlüssellängen in Zukunft als zu unsicher erscheinen lassen.

› Kryptoalgorithmen

Es gibt zwei Arten von Kryptoalgorithmen mit Schlüsseln: symmetrische Algorithmen und Algorithmen mit öffentlichen Schlüsseln (Public-Key-Algorithmen).

Bei symmetrischen Algorithmen sind Chiffrierschlüssel und Dechiffrierschlüssel entweder identisch, oder der Dechiffrierschlüssel lässt sich aus dem Chiffrierschlüssel berechnen und umgekehrt. Es gilt:

$$E_K(M) = C$$

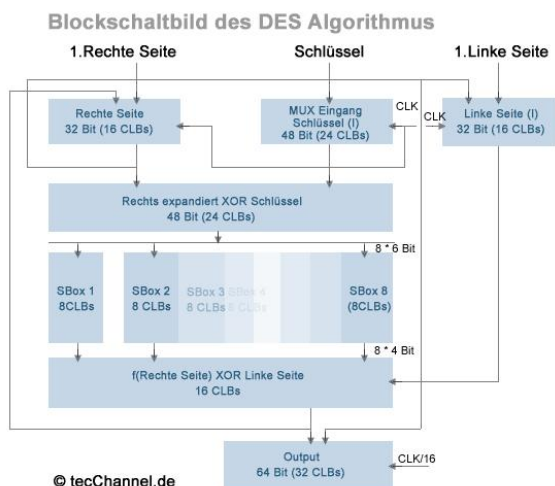
$$D_K(C) = M$$

M = Klartext Nachricht
C = Chiffretext (verschl. Nachricht)
E = Verschlüsselungsfunktion
D = Entschlüsselungsfunktion
K = Schlüssel

Bei symmetrischen Algorithmen benutzen Sender (oft als Alice bezeichnet) und Empfänger (namentlich Bob) einen gemeinsamen (geheimen) Schlüssel. Dieser geheime Schlüssel muss vor Beginn der verschlüsselten Kommunikation auf eine sichere Weise vereinbart und ausgetauscht worden sein. Zum Beispiel, indem sich Alice und Bob getroffen haben.

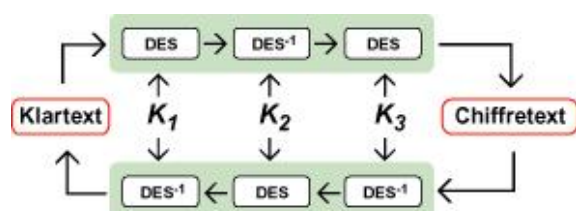
› **DES**

Das bekannteste und am weitesten verbreitete symmetrische Verschlüsselungsverfahren ist der Data Encryption Standard (DES). Es wurde 1976 in den Vereinigten Staaten als Bundesstandard anerkannt und benutzt eine Schlüssellänge von 56 Bit.



DES: Blockschaltbild des DES-Algorithmus

DES ist auf Standardrechnern in Wochen bis Monaten zu knacken. Anfang 1999 war es möglich, durch die Nutzung der Leerlaufzeit vieler per Internet verbundener Computer, eine per DES verschlüsselte Nachricht innerhalb von 23 Stunden zu dechiffrieren. Erreicht wurde dies einfach durch das Ausprobieren aller möglichen Schlüssel. Spezialrechner brauchen für die gleiche Aufgabe nur einen Bruchteil dieser Zeit. Eine auch heute noch sichere Variante von DES ist Triple-DES, das heißt, die dreimalige, hintereinander geschaltete Anwendung von DES. Die Schlüssellänge steigt dadurch auf 168 Bit.



Dreifach sicher: Beim Triple-DES wird gleich drei Mal verschlüsselt.

Derzeit läuft eine Ausschreibung des NIST für den Advanced Encryption Standard (AES), den Nachfolger von DES. Bei AES soll es sich um einen frei verfügbaren symmetrischen 128-Bit-Blockchiffre handeln. Als Schlüsselgrößen sind 128, 192 und 256 Bit gefordert. In der engeren Auswahl sind seit August 1999 noch fünf Algorithmen, nämlich MARS, RC5, RIJNDAEL, Serpent und Twofish.

› Public-Key-Algorithmen

Algorithmen mit öffentlichem Schlüssel beruhen auf der Tatsache, dass manche Dinge im Leben einfach auszuführen, aber nur schwer rückgängig zu machen sind. Eine Vase aus zehn Metern Höhe fallen zu lassen, bereitet keine große Mühe; aus den Scherben die Vase wieder zusammenzukleben, ist jedoch fast unmöglich.

Bei den Zahlen gibt es ähnliche Phänomene: Zahlen miteinander zu multiplizieren - selbst sehr große - ist leicht. Aber ein Produkt in seine (unbekannten) Faktoren zu zerlegen, ist vergleichsweise schwer.

Bei dem Public-Key-Verfahren wird jeweils vom Sender ein Schlüssel zur Chiffrierung und vom Empfänger ein anderer, zugehöriger Schlüssel für die Dechiffrierung verwendet. Sender und Empfänger verwenden *Schlüsselpaare*. Bei einem guten asymmetrischen Verfahren kann trotz der Kenntnis eines Schlüssels der andere nicht abgeleitet werden. Es gilt:

$$E_{pK}(M) = pC$$

$$D_{sK}(pC) = M$$

$$E_{sK}(M) = sC$$

$$D_{pK}(sC) = M$$

pC = mit öffentlichem Schlüssel verschlüsselte Nachricht

sC = mit privatem Schlüssel verschlüsselte Nachricht

pK = öffentlicher Schlüssel

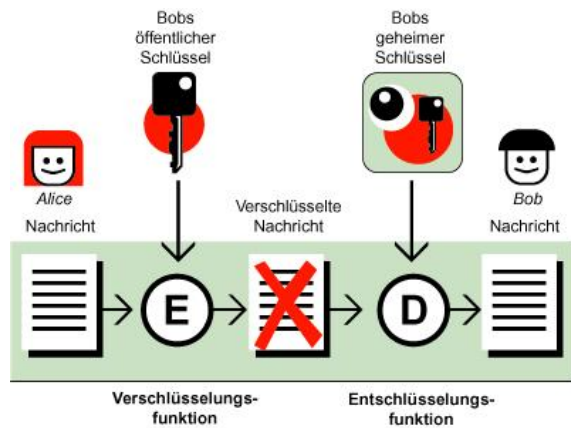
sK = privater Schlüssel

Zusammengefasst haben Public-Key-Verfahren folgende Merkmale:

- › Jeder potenzielle Kommunikationsteilnehmer besitzt einen öffentlichen Schlüssel (Public Key) und einen persönlichen Schlüssel (Private Key).
- › Der Public Key darf öffentlich bekannt sein, der Private Key muss geheim gehalten werden.
- › Es ist (praktisch) unmöglich, aus dem Public Key den Private Key zu berechnen.
- › Der Sender einer vertraulichen Nachricht muss den Public Key des Empfängers kennen.

› Funktionsweise Public Key

Will Alice eine geheime Nachricht an Bob schicken, verschlüsselt sie die Nachricht mit Bobs öffentlichem Schlüssel. So kann nur Bob diese Nachricht mit seinem privaten Schlüssel (Private Key) wieder entschlüsseln. Es ist dadurch möglich, dass Alice an Bob verschlüsselte Nachrichten schickt, ohne dass die beiden zuvor über einen sicheren Kanal einen gemeinsamen Schlüssel vereinbaren mussten. Das ist der große Vorteil gegenüber den symmetrischen Verfahren.



Getrennt: Beim Verfahren mit Public Keys kommen zwei verschiedene Schlüssel zum Einsatz.

Bei diesem Verfahren spielt es keine Rolle, wer sonst noch Bobs öffentlichen Schlüssel kennt. Eine einmal mit diesem Schlüssel verschlüsselte Nachricht kann nur noch mit Bobs privatem Schlüssel wieder gelesen werden.

› Primfaktorzerlegung

Die Zerlegung einer (großen) Zahl in ihre Primfaktoren ist eines der ältesten Probleme der Zahlentheorie. Neben der versuchsweisen Division, die einfach, aber bei großen Zahlen sehr zeitaufwendig ist, existieren einige effizientere Faktorisierungsalgorithmen, von denen an dieser Stelle nur der neueste und vermutlich bald auch schnellste, das so genannte Zahlenkörpersieb (Number Field Sieve, NFS) genannt werden soll.

Im Moment ist man in der Lage, Zahlen mit etwa 130 Dezimalstellen (entspricht zirka 440 Bit) zu faktorisieren. 1993 benötigte man dazu etwa 5000 MIPS-Jahre (eine theoretische Größe für den Rechenaufwand). Vor einiger Zeit schaltete man über das Internet 1600 Rechner zusammen, die gemeinsam etwa acht Monate brauchten. Nach Aussage der beteiligten Wissenschaftler wäre der Aufwand unter Verwendung des neuen NFS nur ein Zehntel dieser Zeit gewesen. Der Rechenaufwand zur Faktorisierung einer großen Zahl mit n Stellen mit Hilfe des NFS (seine heuristische, asymptotische Laufzeit) lässt sich mit der folgenden Formel abschätzen:

$$e^{(1,923+O(1))(\ln(n))^{1/3} (\ln(\ln(n)))^{2/3}}$$

Wie man sieht, steigt der Rechenaufwand mit der Länge der zu faktorisierenden Zahl exponentiell an. Die Primfaktorzerlegung großer Zahlen ist ein seit langem sehr intensiv untersuchtes Problem. Große Fortschritte in Form neuer, wesentlich schnellerer Algorithmen sind daher in diesem Bereich unwahrscheinlich. Dies macht Kryptoalgorithmen, die auf der Primfaktorzerlegung beruhen, zu guten Kandidaten für sichere Kryptoalgorithmen.

› Das BSI empfiehlt

Bei asymmetrischen Verschlüsselungsverfahren haben sich inzwischen eine Reihe unterschiedlicher Methoden etabliert. Vor allem zwei gelten derzeit als sicher und werden unter anderem vom Bundesamt für Sicherheit in der Informationstechnik (BSI (<http://www.bsi.bund.de/aufgaben/projekte/pbdigsig/index.htm>)) empfohlen:

1. ElGamal, 1985

Das Prinzip des Algorithmus von ElGamal zur asymmetrischen Verschlüsselung beruht auf der Schwierigkeit "diskrete Logarithmen modulo einer Primzahl" zu berechnen. In

praktischen Anwendungen variiert die Primzahl zwischen 512 Bits (geringe Sicherheit) und 1024 Bits (sehr hohe Sicherheit). Eine Variante des ElGamal-Verfahrens ist der 1991 vom National Institute of Standards and Technology publizierte Digital Signature Standard (DSS), der den Digital Signature Algorithm (DAS) spezifiziert. Der ElGamal-Algorithmus ist nicht patentiert.

2. RSA (Rivest, Shamir, Adleman), 1977

RSA, benannt nach den Entwicklern Rivest, Shamir, Adleman, ist das bekannteste Public-Key-Verfahren, die am weitesten verbreitete asymmetrische Verschlüsselungsmethode und ein Quasi-Standard im Internet. Das Prinzip beruht auf der Schwierigkeit, große natürliche Zahlen in der Größenordnung $10^{\text{hoch } 150}$ (beispielsweise 200 Dezimalstellen oder 665 Bits) in ihre Primfaktoren zu zerlegen. In praktischen Anwendungen variieren die Zahlen zwischen 512 Bits (geringe Sicherheit) und 2048 Bits (sehr hohe Sicherheit). RSA ist weltweit seit Ende 2000 frei von Patenten.

› RSA

Bei RSA beruhen öffentlicher und privater Schlüssel auf einem Paar sehr großer Primzahlen (100 bis 200 Stellen und mehr). Es wird allgemein angenommen, dass der Aufwand zur Wiederherstellung des Klartextes aus dem Chiffretext und dem öffentlichen Schlüssel äquivalent zur Faktorisierung des Produktes der beiden Primzahlen ist. (Dies ist allerdings streng genommen nur eine qualifizierte Vermutung, es wurde nie bewiesen, dass es wirklich so ist.)

RSA ist um den Faktor 100 bis 1000 langsamer als DES. Dies mag im ersten Moment wie ein Nachteil von RSA aussehen, ist aber tatsächlich eher von Vorteil. Denn für die Ver- und Entschlüsselung von normalen Mitteilungen fällt diese Zeit praktisch nicht ins Gewicht. Wer aber RSA mittels einem Brute-Force-Angriff, also dem Ausprobieren aller möglichen Schlüssel, brechen möchte, tut sich umso schwerer, je langsamer der Algorithmus ist.

Es ist zurzeit möglich, einen 512-Bit-langen RSA-Schlüssel zu knacken. Der Aufwand hierfür beträgt im Moment etwa 8000 MIPS-Jahre. Schlüssellängen von 1024 Bit oder gar 2048 Bit sind bei RSA nach menschlichem Ermessen in nicht absehbarer Zukunft absolut sicher.

› Schlüsselgenerierung bei RSA

1. Schritt: Der Sender, A, wählt zwei große Primzahlen **p** und **q**. A berechnet das Produkt **n = p * q**. Wichtig: p und q müssen sich in ihrer Länge deutlich unterscheiden. Andernfalls könnten sie aus n leicht bestimmt werden, indem in der Umgebung von (Wurzel aus n) alle Primzahlen getestet werden.

2. Schritt: A wählt seinen öffentlichen Schlüssel **e** so, dass e und **(p-1) * (q-1)** keinen gemeinsamen Primfaktor außer der 1 haben.

3. Schritt: A berechnet seinen privaten Schlüssel **d** mit Hilfe der Formel: **ed = 1 mod ((p-1) * (q - 1))**. Das bedeutet: Zur Berechnung seines privaten Schlüssels ist die Kenntnis von p und q erforderlich, die daher ebenfalls geheim gehalten werden müssen.

Die Zahlen e und n bilden den öffentlichen Schlüssel, d ist der private Schlüssel. Die beiden, zur Generierung der Schlüssel verwendeten Primzahlen, können jetzt verworfen werden, denn sie werden nicht mehr benötigt. Selbstverständlich dürfen sie niemals bekannt gegeben werden.

Zur Verschlüsselung einer Nachricht wird diese in Blöcke zerlegt, deren Länge sich aus der größten Zweierpotenz bestimmt, die kleiner ist, als n Stellen hat. Die Verschlüsselung erfolgt mit:

$$c_i = m_i^e \bmod n \quad \text{für alle Nachrichtenblöcke } m_i$$

Die Entschlüsselung der verschlüsselten Blöcke c_i erfolgt mit

$$m_i = c_i^d \bmod n$$

› Sicherheit von RSA

Wesentlich für die Sicherheit von RSA ist die Auswahl geeigneter Primzahlen. Sie müssen erstens zufällig gewählt werden und zweitens groß genug sein, um bei steigender Rechenleistung auch in zehn oder 20 Jahren noch einer Primfaktorzerlegung standhalten zu können. Sehr leistungsfähige Rechner könnten in den nächsten Jahren die den Einwegfunktionen zu Grunde liegenden Gleichungen umkehren. Man sollte deshalb im Zweifelsfall eine große Schlüssellänge wählen.

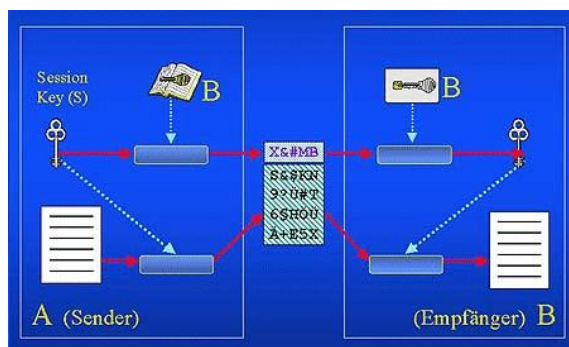
Das BSI verlangt in einem [Papier](http://www.bsi.bund.de/aufgaben/projekte/pbdigsig/index.htm) für den Modulus $n = p * q$ eine Bitlänge von mindestens 2048. Für den Zeitraum bis Ende 2004 genügen noch 1024 Bit.

Compaq hat aktuell eine effizientere Methode für die RSA-Verschlüsselung entwickelt. Beim MultiPrime-Verfahren basiert der geheime Schlüssel nicht wie üblich auf nur zwei, sondern auf drei oder mehr Primzahlen. Dadurch sollen Entschlüsselung und Signatur deutlich schneller und ressourcensparender durchzuführen sein. Den Modulus auf drei Primzahlen zu verteilen bewirkt nach Compaq-Angaben eine theoretische Geschwindigkeitssteigerung um den Faktor 6,7. [RSA Security](http://www.rsasecurity.com) verspricht sich auch in SmartCards zumindest eine Verdopplung der Performance.

Operationen mit den dazu gehörenden öffentlichen Schlüssel (Verschlüsselung und Signaturprüfung) dagegen sind von dem Verfahren nicht betroffen. RSA Security hat MultiPrime in seine Entwicklerkits und Libraries.

› Hybride Verschlüsselung

Da asymmetrische Verschlüsselungssysteme in der Regel sehr viel langsamer arbeiten als symmetrische Algorithmen, werden bei den im Internet gebräuchlichen Verschlüsselungsprogrammen häufig beide Verfahren eingesetzt. Bei einem Verbindungsaufbau wird zunächst mit Hilfe einer asymmetrischen Verschlüsselung ein Sitzungsschlüssel (*Session Key*) gesichert übertragen. Dieser wird anschließend für eine symmetrische Verschlüsselung genutzt. Durch diese Kombination - man spricht von hybrider Verschlüsselung - vereinigt man einen gesicherten, aber langsamen Schlüsseltausch mit einer schnellen, aber weniger sicheren Verschlüsselung.



Das Beste zweier Welten: Hybridverfahren mit asymmetrischer/symmetrischer Verschlüsselung. (Quelle Teletrust)

Das Hybridverfahren läuft wie folgt ab: Der Sender A erzeugt in seiner vertrauenswürdigen Umgebung einen möglichst zufälligen symmetrischen Schlüssel S, den so genannten *Session Key* und kodiert mit diesem seine Nachricht. Diesen Schlüssel selbst chiffriert der Sender mit dem öffentlichen (asymmetrischen) Schlüssel des Empfängers B. Beides, die mit S verschlüsselte Nachricht und der mit dem öffentlichen Schlüssel von B kodierte Sitzungsschlüssel, werden nun an den Empfänger übermittelt.

Da der Empfänger B den Sitzungsschlüssel nicht kennt, muss er zunächst den chiffrierten (symmetrischen) Sitzungsschlüssel entschlüsseln. Dies erfolgt mit seinem geheimen (asymmetrischen) Schlüssel. Den so gewonnenen Sitzungsschlüssel S kann er nun dazu verwenden, die chiffriert übermittelte Nachricht wieder zu dechiffrieren und somit Kenntnis des Nachrichteninhalts zu erlangen.

› Kryptoanalytische und andere Angriffe

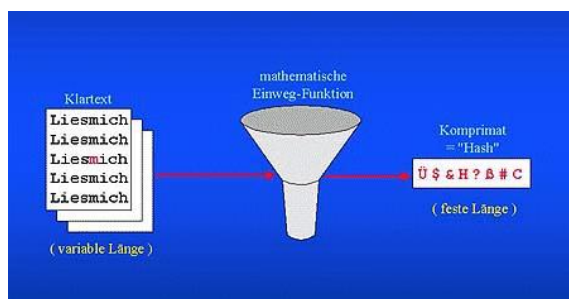
Es gibt verschiedene Möglichkeiten zum Angriff gegen kryptographische Protokolle. Am offensichtlichsten ist der Weg des Brute-Force-Angriffs. Bei ihm werden einfach so lange alle möglichen Schlüssel auf den chiffrierten Text angewandt, bis der lesbare Klartext vorliegt. Daneben gibt es, je nach Algorithmus, mathematisch oft sehr anspruchsvolle Analysemöglichkeiten, die auf bestimmten Eigenheiten des verwendeten Kryptoalgorithmus aufsetzen. Gute Kryptoalgorithmen zeichnen sich dadurch aus, dass der Aufwand für derartige Angriffe genauso groß oder größer als der Aufwand eines Brute-Force-Angriffs ist.

In der Praxis ist jedoch das Risiko, das aus der Ausspähung eines Schlüssels oder dessen Gewinnung durch Bestechung, Erpressung oder Drohung mit Gewalt erwächst, um Größenordnungen höher, als das Risiko, das aus kryptoanalytischen Angriffen resultiert. Die Frage des sicheren Austausches und der absolut sicheren Aufbewahrung von Schlüsseln hat daher eine besondere Bedeutung.

Der sicheren Kommunikation droht noch aus einer anderen Richtung Gefahr. Staatliche Institutionen tun sich noch immer sehr schwer mit der Möglichkeit des Bürgers, unbelauscht zu kommunizieren. Zu sehr haben sich NSA, FBI, die Bundes- und Landeskriminalämter und der Verfassungsschutz daran gewöhnt, jederzeit Zugriff auf alle sie interessierenden Daten der Bürger zu bekommen. Ein Mensch, der Wert darauf legt, seine alltägliche Kommunikation unbelauscht von nationalen und internationalen Organisationen zu praktizieren, gerät leicht in den Verdacht, etwas Verbotenes zu tun. Einschlägige Politiker und Sicherheitsexperten sind dann schnell mit dem Argument zur Hand, wer nichts zu verbergen habe, brauche auch keine Angst vor staatlicher Überwachung zu haben.

› Hash-Funktionen für Signaturen

Verschlüsselungsverfahren wie RSA erreichen nur den Schutz der Vertraulichkeit einer Nachricht. Neben den eigentlichen Signaturverfahren zum Schutz der Vertraulichkeit benötigt man noch eine Methode, den Urheber einer Nachricht beweisbar zu dokumentieren. In der Regel erfolgt dies mit kryptographischen Prüfsummen, so genannten *Hash-Funktionen*. Das sind mathematische Methoden, die aus einem beliebigen Klartext nach einem vorbestimmten Verfahren eine Prüfziffer (Komprimat) generieren. Die Funktion verwandelt einen Klartext so in ein entsprechendes Komprimat um, dass auch die kleinste Veränderung des ursprünglichen Texts zu einer gänzlich anderen Prüfziffer führt.



Nicht umkehrbar: das Ergebnis der Hash-Funktion (Quelle Teletrust)

Es gehört zu den Forderungen an diese mathematische Funktion, dass aus dem einmal erzeugten Komprimat der ursprüngliche Text nicht wieder rekonstruiert werden kann. Eine solche Hash-Funktion ist nicht umkehrbar und gilt somit als Einwegfunktion. Anders als beim Chiffrieren, ist also eine Wiederherstellung des ursprünglichen Textes nicht

möglich.

Außerdem muss die Hash-Funktion möglichst *kollisionsfrei* sein. Verschiedene Nachrichten mit gleichem Hashwert sollen möglichst selten vorkommen. So ist mit einer praktisch vernachlässigbaren Unsicherheit ein bestimmter Hashwert das Ergebnis eines und nur eines ursprünglichen Klartextes.

Der Vorteil dieses Verfahrens liegt in der Tatsache, dass anstatt des gesamten Textes lediglich ein kurzer Hashwert besonders geschützt werden muss.

› Hash-Funktionen in der Praxis

Für digitale Signatur-Verfahren ist die Festlegung auf eine Hash-Funktion notwendig. Verfügbare Einweg-Hash-Funktionen sind:

SHA/SHA-1 (Secure Hash Algorithm One)

SHA wurde von der NSA entwickelt und als US-Standard angenommen. Eine leicht modifizierte Form hat als SHA-1 den Algorithmus inzwischen ersetzt. Der mit SHA-1 erzeugte Hashwert wird für den DSA (Digital Signature Algorithm), der im DSS (Digital Signature Standard) spezifiziert wird, benötigt. Der Hashwert hat eine Länge von 160 Bit.

MD2, MD4, MD5 (Message Digest)

MD4 und MD5 sind Hash-Funktionen, die von R. Rivest (RSA Laboratories) entwickelt und im Zusammenhang mit dem PEM-Standard (Privacy Enhanced Mail) vorgestellt wurden. MD5 ist eine Weiterentwicklung von MD4. Die Algorithmen erzeugen einen Message Digest (Hashwert) von 128 Bit Länge.

RIPED-128, RIPEMD-160 (RIPE-Message Digest)

RIPEMD wurde im Rahmen des EU-Projektes RIPE (RACE Integrity Primitives Evaluation, 1988-1992) ins Leben gerufen. (RIPE-Message Digest). Wegen kryptographischer Schwächen von MD4 und MD5 wurde RIPEMD von Hans Dobbertin, Antoon Bosselaers und Bart Preneel entwickelt. Der Hashwert ist entweder 128 Bit (RIPEMD-128) oder 160 Bit (RIPEMD-160) lang.

RSA Data Security hat auf Grund der Schwächen verfügt, dass MD4 und MD5 für zukünftige Hash-Funktionen nicht implementiert werden sollte. Generell bieten Hash-Funktionen mit längeren Prüfwerten höhere Sicherheit. Daher sollten zukünftig SHA-1 oder RIPEMD-160 verwendet werden. RIPEMD-160 scheint sich in Europa und SHA-1 in den USA als de facto Standard durchzusetzen.

Erst die Kombination aus asymmetrischen Verschlüsselungsverfahren und Hashwerten bietet die Möglichkeit, ein Analogon zur menschlichen Unterschrift zu erzeugen.

› Fazit

Der Mittels der Kryptographie unternommene Versuch, Daten über verschiedenste Verschlüsselungsmethoden geheim zu halten, dient der Absicherung des individuellen Rechts auf Unantastbarkeit der [Privatsphäre](http://www.tecchannel.de/tecvision/191/index.html) (http://www.tecchannel.de/tecvision/191/index.html) . Denn gerade diese gilt es im Datenrausch des Internets zu schützen, sei es bei [Bestellungen](http://www.tecchannel.de/internet/394/index.html) (http://www.tecchannel.de/internet/394/index.html) , [Bankgeschäften](http://www.tecchannel.de/internet/62/index.html) (http://www.tecchannel.de/internet/62/index.html) , beim [normalen E-Mail-Verkehr](http://www.tecchannel.de/internet/398/index.html) (http://www.tecchannel.de/internet/398/index.html) oder beim generellen [Surfen](http://www.tecchannel.de/internet/395/index.html) (http://www.tecchannel.de/internet/395/index.html) durch das Internet mittels entsprechender [Absicherungsmethoden](http://www.tecchannel.de/internet/284/index.html) (http://www.tecchannel.de/internet/284/index.html) .

Wie jeher, zeigen sich zwei Seiten - einerseits der Versuch, Daten zu schützen und damit einhergehend Versuche, diesen Schutz aufzubrechen. Allerdings tragen Aufklärungsversuche in den Medien langsam Früchte.

Wachsende Sensibilisierung der Bürger, mehr aber noch der massive Druck der Industrie, die ein vitales Interesse an sicherer Kommunikation hat, haben im vergangenen

Jahr zu einer deutlichen Liberalisierung auf diesem Gebiet geführt. Das am meisten verbreitete Programm zur sicheren Verschlüsselung von E-Mails, **PGP** (<http://www.pgpi.org/>) (Pretty Good Privacy), durfte jahrelang nicht aus Amerika exportiert werden, weil es militärisch(!) sensible Technologie enthielt. Nur über den Umweg, den gedruckten Sourcecode von USA nach Europa zu schaffen, hier wieder einzuscannen und neu zu übersetzen, war es möglich, PGP auch im Rest der Welt zu nutzen. Dieser absurde Zustand ist inzwischen glücklicherweise beseitigt. Trotzdem muss auch weiterhin sorgsam darauf geachtet werden, dass staatliche Behörden sich nicht erneut Hintertürchen schaffen, um die Bevölkerung oder die Wirtschaft zu belauschen. (sda/mha)

Wozu man diese kryptographischen Verfahren benötigt, lesen Sie in einem gesonderten Beitrag zur **digitalen Signatur** (<http://www.tecchannel.de/internet/402/index.html>), die bald auch virtuelle Behördengänge über das Internet ermöglichen soll.

Literatur:

A.K. Lenstra and H.W. Lenstra, Jr., eds., Lecture Notes in Mathematics 1554: The Development of the Number Field Sieve, Springer-Verlag, 1993

C.E. Shannon, Collected Papers: Claude Elmwood Shannon, N.J.A. Sloane and A.D. Wyner, eds., New York: IEEE Press, 1993

Schneier, Bruce (1996): Angewandte Kryptographie: Protokolle, Algorithmen und Sourcecode in C; Addison-Wesley

› Weitere Themen zu diesem Artikel:

[Elektronisch unterschreiben](http://www.tecchannel.de/internet/402/index.html) (<http://www.tecchannel.de/internet/402/index.html>)

[Ist Privatsphäre noch möglich?](http://www.tecchannel.de/tecvision/191/index.html) (<http://www.tecchannel.de/tecvision/191/index.html>)

[Dem Surfer auf der Spur](http://www.tecchannel.de/internet/284/index.html) (<http://www.tecchannel.de/internet/284/index.html>)

[Sichere E-Mail](http://www.tecchannel.de/internet/398/index.html) (<http://www.tecchannel.de/internet/398/index.html>)

[Safer Surfen](http://www.tecchannel.de/internet/395/index.html) (<http://www.tecchannel.de/internet/395/index.html>)

[Versteckter Schutz gegen Datenraub](http://www.tecchannel.de/multimedia/377/index.html) (<http://www.tecchannel.de/multimedia/377/index.html>)

[Bezahlen im Internet](http://www.tecchannel.de/internet/394/index.html) (<http://www.tecchannel.de/internet/394/index.html>)

[HBCI - Der neue Homebanking-Standard](http://www.tecchannel.de/internet/62/index.html) (<http://www.tecchannel.de/internet/62/index.html>)

Copyright © 2001
IDG Interactive GmbH

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium in Teilen oder als Ganzes bedarf der schriftlichen Zustimmung der IDG Interactive GmbH. DPA-Texte und Bilder sind urheberrechtlich geschützt und dürfen weder reproduziert noch wiederverwendet oder für gewerbliche Zwecke verwendet werden. Für den Fall, dass in tecChannel unzutreffende Informationen veröffentlicht oder in Programmen oder Datenbanken Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht. Die Redaktion übernimmt keine Haftung für unverlangt eingesandte Manuskripte, Fotos und Illustrationen. Für Inhalte externer Seiten, auf die von tecChannel aus gelinkt wird, übernimmt die IDG Interactive GmbH keine Verantwortung.