C61_Sicherheitsprobleme

Bei der Datenübertragung wird unter anderem auf folgende Schutzziele Wert gelegt:

Authentizität

Eine Datenübertragung kommt wirklich von dem angegebenen Sender.
Verwirklicht durch: **Zertifikate.**

Vertraulichkeit

Eine Datenübertragung ist von keinem außer dem berechtigten Empfänger gelesen worden.
Auch als Geheimhaltung bezeichnet.
Verwirklicht durch: **Verschlüsselung**.

Integrität

Eine Datenübertragung ist nicht verändert worden.
Verwirklicht durch: **Prüfung von Hashwerten**.

Nichtabstreitbarkeit

Der Sender kann beweisen, dass der Empfänger die übermittelten Daten bekommen hat bzw. der Empfänger kann den Empfang nicht abstreiten. Ebenso kann der Sender nicht abstreiten, dass er die Nachricht mit diesem Inhalt an den Empfänger geschickt hat.

Verschlüsselung

Verschlüsselung nennt man den Vorgang, bei dem ein „Klartext“ mit Hilfe eines Verschlüsselungsverfahrens in einen „Geheimtext“ umgewandelt wird. Als Parameter des Verschlüsselungsverfahrens werden ein oder mehrere Schlüssel verwendet. Das Forschungsgebiet, das sich mit der Verschlüsselung und ihrer Geschichte beschäftigt, wird als Kryptografie bezeichnet.

Den umgekehrten Vorgang, also die Verwandlung des Geheimtextes zurück in den Klartext, nennt man Entschlüsselung. Die Algorithmen zur Ver- bzw. Entschlüsselung müssen nicht identisch sein. Ebenso wenig müssen identische Schlüssel zum Einsatz kommen. Das Forschungsgebiet der Entschlüsselung heißt Kryptoanalyse und ist natürlich eng verwandt mit der Kryptografie.

Eine grobe Unterscheidung in symmetrische und asymmetrische Verschlüsselungssysteme ergibt sich aus der Weise, wie kryptografische Schlüssel an die am Verfahren Beteiligten vermittelt werden:

Bei symmetrischen Systemen besitzen beide Kommunikationspartner denselben Schlüssel und müssen diesen vor Beginn der Kommunikation sicher ausgetauscht haben (z.B. mittels Diffie-Hellman-Schlüsselaustausch (siehe unten) oder der Zusendung per Post). Ein bekanntes klassisches symmetrische Verfahren ist DES (komplexitätstheoretisch sicher). Zu den modernen und derzeit als sicher angesehenen Verfahren gehören der Rijndael, Twofish sowie 3DES, wobei dem Rijndael durch seine Erhebung zum Advanced Encryption Standard und aufgrund seiner Bevorzugung durch staatliche US-amerikanische Stellen eine herausragende Rolle zukommt.

Asymmetrische Systeme zeichnen sich dadurch aus, dass für jeden Teilnehmer ein Schlüsselpaar generiert wird. Ein Schlüssel jedes Paars wird veröffentlicht, der andere bleibt geheim. Die Asymmetrie ergibt sich, weil ein Schlüssel eines Paars immer nur ver- und der

andere immer nur entschlüsseln können. Das bekannteste dieser Verfahren ist das RSA-Kryptosystem.

Diffie-Hellman-Schlüsselaustausch

Der **Diffie-Hellman-Schlüsselaustausch** beschreibt eine Möglichkeit, kryptografische Schlüssel *sicher* über *unsichere* Kanäle auszutauschen. Der Diffie-Hellman-Schlüsselaustausch ist aber *kein* Verschlüsselungsverfahren.

Das Diffie-Hellman-Verfahren setzte der langen Tradition von Streichlisten und Codebüchern ein Ende. Noch während des Zweiten Weltkrieges mussten die Benutzer der ausgeklügelten Verschlüsselungsmaschinen (zum Beispiel die Enigma) Codebücher mit sich führen, um für jeden einzelnen Tag des Jahres zu wissen, welchen Schlüssel der Sender verwendet. Wird ein solches Codebuch geraubt, ist die Verschlüsselung hinfällig geworden. Besonders beim Militär waren die Zuteilung und der Transport solcher hochgeheimer Codebücher stets das größte Sorgenkind.

Beim Diffie-Hellman-Verfahren werden Eigenschaften diskreter Logarithmen ausgenutzt. Es ist nicht zuletzt die Grundlage für das Elgamal-Kryptosystem. Das Verfahren dient nicht zum Austausch eines vorher erzeugten Schlüssels, sondern ermöglicht den Kommunikationspartnern, auf Basis der ausgetauschten Informationen denselben Schlüssel zu generieren.

Geschichte

Der Algorithmus wurde von Martin Hellman gemeinsam mit Whitfield Diffie und Ralph Merkle an der Stanford-Universität (Kalifornien) entwickelt und 1976 veröffentlicht.

Wie erst 1997 bekannt wurde, hatte das britische *Government Communications Headquarters* (GCHQ) schon in den 1960er Jahren den Auftrag erteilt, aufgrund der hohen Kosten bei der damals üblichen Schlüsselverteilung einen anderen Weg für die Schlüsselverteilung zu finden.

Die von James Ellis, Clifford Cocks und Malcolm Williamson geäußerten Ideen ähnelten dem Diffie-Hellmann-Verfahren. Das GCHQ hat einerseits wegen der Geheimhaltung, andererseits wegen des für die Briten aus Sicht der frühen 1970er Jahre fraglichen Nutzens nie ein Patent beantragt.

Funktionsweise

Zwei Kommunikationspartner wollen über ein unsicheres Medium, etwa eine Kabel- oder Funkleitung, verschlüsselt kommunizieren. Sie wollen ein symmetrisches Kryptosystem einsetzen. Dazu benötigen beide jedoch zunächst einen Schlüssel, den sie über den Diffie-Hellman-Schlüsselaustausch vereinbaren.

1. Die Kommunikationspartner einigen sich zunächst auf eine Primzahl p und eine Primitivwurzel $g \bmod p$ mit $2 \leq g \leq p-2$. Diese Parameter müssen nicht geheim bleiben, können also insbesondere auch über ein unsicheres Medium übertragen werden.
2. Beide Kommunikationspartner erzeugen jeweils eine geheim zu haltende Zufallszahl a bzw. b aus der Menge $\{1, \ldots, p-2\}$. a und b werden nicht übertragen, bleiben

also dem jeweiligen Kommunikationspartner, aber auch potenziellen Lauschern unbekannt.
3. Die Kommunikationspartner berechnen $A = g^a \bmod p$ bzw. $B = g^b \bmod p$. Nun werden A und B über das unsichere Medium übertragen.
4. Die Kommunikationspartner berechnen nun $B^a \bmod p = (g^b \bmod p)^a \bmod p = (g^{ba}) \bmod p = (g^{ab}) \bmod p = (g^a \bmod p)^b \bmod p = A^b \bmod p = K$. Das Ergebnis K ist für beide Partner gleich und kann als Schlüssel für die weitere Kommunikation verwendet werden.

Beispiel

Die Kommunikationspartner seien Alice und Bob. Das Beispiel benutzt sehr kleine Zahlen. In der tatsächlichen Anwendung werden Zahlen mit mehreren Hundert Dezimalstellen benutzt.

1. Alice und Bob einigen sich auf $p = 13$ und $g = 2$.
2. Alice wählt die Zufallszahl $a = 5$. Bob wählt die Zufallszahl $b = 7$.
3. Alice berechnet $A = 2^5 \bmod 13 = 6$ und sendet dieses Ergebnis an Bob.
4. Bob berechnet $B = 2^7 \bmod 13 = 11$ und sendet dieses Ergebnis an Alice.
5. Alice berechnet $K = 11^5 \bmod 13 = 7$.
6. Bob berechnet $K = 6^7 \bmod 13 = 7$.
7. Beide erhalten das gleiche Ergebnis $K = 7$.

Ein eventuell vorhandener Lauscher könnte zwar die Zahlen 13, 2, 6 und 11 mithören, das eigentliche gemeinsame Geheimnis von Alice und Bob $K = 7$ bleibt ihm aber verborgen.

Sicherheit

Die Sicherheit des Verfahrens basiert auf der Verwendung einer sogenannten Einwegfunktion. Hierbei macht man sich die Tatsache zunutze, dass es zwar sehr einfach ist, eine Zahl zu potenzieren, dass es jedoch nur mit sehr großem Aufwand möglich ist, den diskreten Logarithmus einer Zahl zu berechnen. In der Praxis werden sehr große Primzahlen verwendet, die Sicherheit kann weiter erhöht werden, wenn $g = (p - 1) / 2$ ebenfalls eine Primzahl ist (g ist dann eine Sophie-Germain-Primzahl).

Potenzielle Lauscher erfahren zwar p, g, A und B. a und b bleiben hingegen unbekannt.

Das Diffie-Hellman-Problem, $K = g^{ab} \bmod p$ zu berechnen, ist lösbar, wenn man diskrete Logarithmen mod p berechnen kann. Eine andere Lösungsmethode ist nicht bekannt.

Das Verfahren gilt als sicher gegen passives Abhören ("Eavesdropping"), da die im Klartext übertragenen Informationen nicht zur Konstruktion des Schlüssels K ausreichen. Wenn der Angreifer jedoch aktiv Datenpakete abfangen und verändert weiterschicken kann, besteht die Möglichkeit des Man-In-The-Middle-Angriffs: Der Angreifer M vereinbart mit Partner A den Schlüssel K_{AM} und mit Partner B den Schlüssel K_{MB}, wobei A und B davon ausgehen, der Austausch fände mit dem berechtigten Kommunikationspartner statt. Auch die folgende symmetrisch verschlüsselte Kommunikation wird über den Angreifer M umgeleitet. Daten von A an B werden von A mit K_{AM} verschlüsselt und können von M gelesen werden, bevor sie mit K_{MB} verschlüsselt an B weitergeschickt werden (und umgekehrt). Dabei kann der Nachrichteninhalt sogar unbemerkt verändert werden. Diesem Problem kann durch elektronische Unterschrift und ähnliche Maßnahmen erfolgreich begegnet werden.

Literatur

- Johannes Buchmann: *Einführung in die Kryptographie*, Springer 2003, ISBN 3-540-40508-9
- Simon Singh: *Geheime Botschaften*, dtv 2001, ISBN 3-423-33071-6

Weblinks

- Beispiel für den Diffie-Hellman-Algorithmus
- RFC 2631 – E. Rescorla: *Diffie-Hellman Key Agreement Method*, Juni 1999. (englisch)
- Diffie-Hellman explained visually

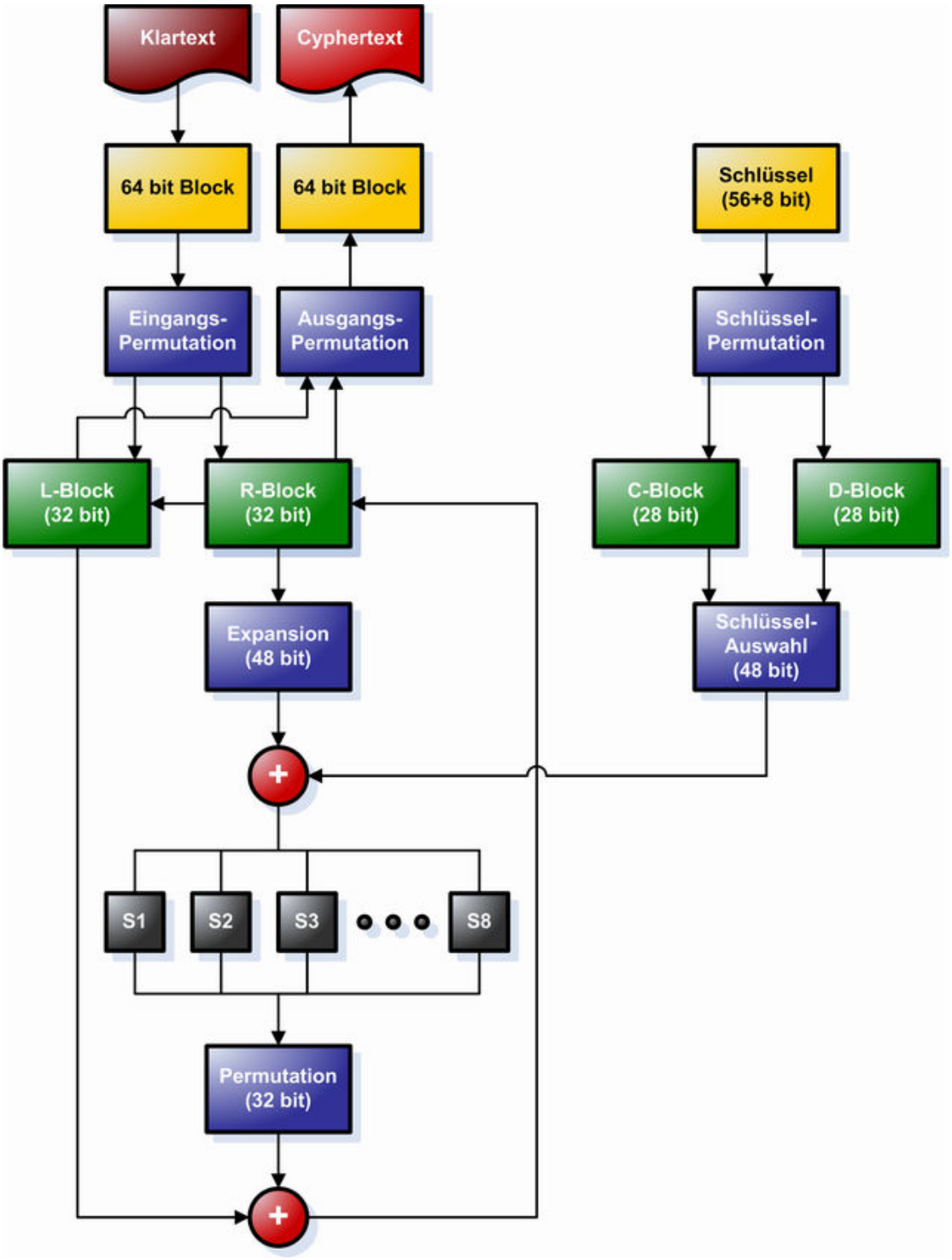
Data Encryption Standard

Der **Data Encryption Standard** (Abkürzung:**DES**) ist ein weit verbreiteter symmetrischer Verschlüsselungsalgorithmus.

Der DES-Algorithmus wurde als offizieller Standard für die US-Regierung im Jahr 1976 bestätigt und wird seither international vielfach eingesetzt. Seine Entstehungsgeschichte hat wegen der Beteiligung der NSA am Design des Algorithmus immer wieder Anlass zu Spekulationen über seine Sicherheit gegeben. Heute wird DES aufgrund der verwendeten Schlüssellänge von nur 56 Bit für viele Anwendungen als nicht ausreichend sicher erachtet.

Die Schlüssellänge kann durch Mehrfachanwendung des DES jedoch leicht vergrößert werden. Als Triple-DES, TDES oder 3DES wird der DES weiterhin am häufigsten, z.B. von Banken in Chipkartenanwendungen, eingesetzt, obwohl der TDES als offizieller Standard für die USA durch den Advanced Encryption Standard (AES) abgelöst wurde.

Bei DES handelt es sich um einen symmetrischen Algorithmus, das heißt zur Ver- und Entschlüsselung wird derselbe Schlüssel verwendet. DES funktioniert als Blockchiffre, das heißt jeder Block wird unter Verwendung des Schlüssels einzeln chiffriert, wobei die Daten in 16 Iterationen beziehungsweise Runden von Substitutionen und Transpositionen (Permutation) nach dem Schema von Feistel "verwürfelt" werden. Die Blockgröße beträgt 64 Bits, das heißt ein 64-Bit-Block Klartext wird in einen 64-Bit-Block Chiffretext transformiert. Auch der Schlüssel, der diese Transformation kontrolliert, besitzt 64 Bits. Jedoch stehen dem Benutzer von diesen 64 Bits nur 56 Bits zur Verfügung; die übrigen 8 Bits (jeweils ein Bit aus jedem Byte) werden zum Paritäts-Check benötigt. Die wirkliche Schlüssellänge beträgt daher nur 56 Bits. Die Entschlüsselung wird mit dem gleichen Algorithmus durchgeführt wobei die einzelnen Rundenschlüssel in umgekehrter Reihenfolge verwendet werden.



Der DES Algorithmus beschreibt zunächst nur wie ein Datenblock mit 64 Bit verarbeitet wird. Zur Verarbeitung einer Nachricht beliebiger Länge gibt es verschiedene so genannte Betriebsmodi ECB, CBC, CFB, OFB. Diese Verfahren lassen sich auf jede Blockchiffre anwenden.

Die Nachricht wird in 64 Bit-Blöcke zerlegt. Beim ECB und CBC wird der letzte Block auf volle 64 Bit ergänzt bzw. ein weiterer Block angefügt. Zum Auffüllen wird die Bitfolge 1000... verwendet. Diese Paddingvorschrift ist eindeutig umkehrbar, d.h. die Füllbits können nach dem Entschlüsseln wieder entfernt werden.

ECB - Electronic Code Book. Die Blöcke werden jeweils mit dem gleichen Schlüssel unabhängig von einander verschlüsselt.

CBC - Cipher Block Chaining Mode. Vor der Verschlüsselung wird der Klartextblock mit dem zuvor verschlüsselten Block verknüpft (XOR-Verknüpfung). Der erste Klartextblock wird dem *Initialisierungsvektor* IV verknüpft. Der IV wird zur Entschlüsslung benötigt, muss jedoch nicht geheim gehalten werden. Dadurch entsteht eine Verkettung (engl. *chain*) der Blöcke. Trotz dieser Verkettung werden bei einem Übertragungsfehler in einem Chiffrenblock nur der betroffene Block und der nachfolgende falsch entschlüsselt. Fehler/Manipulationen in einem Block wirken sich also bei der Entschlüsselung nur auf den fehlerhaften/manipulierten und den nächsten Block aus. Trotzdem gilt, dass durch eine einzige Änderung im Klartext sich alle nachfolgenden Blöcke in der Folge stark verändern. Ein Angreifer kann daher die Chiffre kaum unbemerkt verändern, ohne dass danach 2 Blöcke erkennbar manipuliert sind. In keinem Fall sind gezielte Veränderungen ohne Kenntnis des Schlüssels möglich. Durch Verwendung eines verschlüsselten Prüfwerts (z.B. CRC) wird dies mit höchster Sicherheit ausgeschlossen.

OFB - Output Feedback Mode. In diesem Modus wird eine Zufallsfolge erzeugt. Dazu wird ein *Initialisierungsvektor* verschlüsselt, das Ergebnis erneut verschlüsselt etc. Die Verschüsselung erfolgt durch XOR-Verknüpfung der Zufallsfolge mit dem Klartext. Der IV darf nur einmalig verwendet werden. Diese Verfahren entspricht einer Stromchiffre vergleichbar zu RC4.

CFB - Cipher Feedback Mode. Hier wird wie beim OFB eine Zufallsfolge erzeugt und zur Verschlüsselung per XOR-Verknüpfung genutzt. Jedoch wird die Chiffre zur Berechnung der Zufallsfolge verwendet (daher die Bezeichnung Cipher Feedback).

RSA-Kryptosystem

RSA ist ein Verfahren oder Algorithmus zur Verschlüsselung elektronischer Daten, der verschiedene Schlüssel zum Ver- und Entschlüsseln verwendet, wobei der Schlüssel zum Entschlüsseln nicht oder nur mit hohem Aufwand aus dem Schlüssel zum Verschlüsseln berechenbar ist. Der Schlüssel zur Verschlüsselung kann daher veröffentlicht werden. Solche Verfahren werden als asymmetrische oder Public Key-Verfahren bezeichnet. Er ist nach seinen Erfindern Ronald L. **R**ivest, Adi **S**hamir und Leonard **A**dleman benannt.

Schüsselverteilung

Der Vorteil asymmetrischer Verfahren besteht in der Vereinfachung der Schüsselverteilung. Bei symmetrischen Verfahren ist der Austausch eines geheimen Schlüssels erforderlich, den Sender und Empfänger gemeinsam nutzen. Der Austausch muss dabei sicher erfolgen. Dies bedeutet der Austausch muss abhörsicher sein. Es ist jedoch auch sicher zu stellen, dass der Schlüssel tatsächlich von der Person stammt, mit der geheime Nachrichten ausgetauscht werden sollen.

Durch Public Key-Systeme entfällt die Notwendigkeit den Schlüsselaustausch gegen Abhören zu härten, da nur der öffentliche Schlüssel ausgetauscht werden muss. Es bleibt aber ein wesentliches Problem, da weiterhin sicherzustellen ist, dass der öffentliche Schlüssel tatsächlich von der Person stammt, mit der geheime Botschaften ausgetauscht werden sollen. In Praxis ist dies oft nur dann möglich, wenn zugleich auch die Geheimhaltung gewährleistet werden kann.

Falls die Kommunikation zwar nicht abhörsicher ist, aber kein Zweifel über die Identität des Partners besteht und die Nachricht nicht von einem Dritten verändert werden kann, ist der Schlüsselaustausch beim Public Key Verfahren problemlos. Allerdings ist dies auch mit anderen Verfahren möglich.

Dazu kann der Schlüssel aus vielen Teilschlüsseln zusammengesetzt werden. Eine Hashfunktion wie SHA-1 erlaubt es aus vielen längeren Teilschlüsseln ein Komprimat zu errechnen, welches als Schlüssel für die symmetrische Verschlüsselung benutzt werden kann. Die Teilschlüssel können zu unterschiedlichen Zeiten und mit höchst unterschiedlichen Methoden ausgetauscht werden.

Hybride Verfahren

RSA ist im Vergleich zu 3DES, AES und SHA-1 um mindestens einen Faktor 1.000 langsamer. Für die Verschlüsselung größerer Datenmengen werden daher symmetrische Verfahren eingesetzt. Es genügt jedoch RSA zum Austausch eines Schlüssels für die symmetrische Verschlüsselung zu nutzen. Entsprechend genügt es beim Signaturverfahren einen SHA-1-Wert statt der gesamten Nachricht zu signieren. Es werden daher praktisch immer hybride Verfahren eingesetzt.

Verfahren

Das Verfahren wurde 1977 entwickelt und basiert auf dem aktuellen Wissensstand, das die Faktorisierung einer großen Zahl, also ihre Zerlegung in ihre Primfaktoren, eine sehr aufwändige Angelegenheit ist, während das Erzeugen einer Zahl durch Multiplikation zweier Primzahlen recht einfach ist. Wenn nun eine Nachricht einem Empfänger verschlüsselt zugeleitet werden soll, generiert dieser einen öffentlichen Schlüssel. Der Nachrichtenabsender verwendet diesen öffentlich bekanntgemachten Schlüssel, indem er damit seine Botschaft verschlüsselt. Nur der Empfänger kann diese "dekodieren", da nur er die "Zusammensetzung" des von ihm erzeugten (öffentlichen) Schlüssels kennt.

Einwegfunktionen

Funktionen wie die Multiplikation/Faktorisierung, bei denen eine Richtung leicht, die andere schwierig zu berechnen ist, bezeichnet man als *Einwegfunktionen* (engl. *one way function*). Um allerdings die Entschlüsselung tatsächlich möglich zu machen, muss es sich um *Falltürfunktionen* (engl. *trap door function*) handeln, die mit Hilfe einer Zusatzinformation auch rückwärts leicht zu berechnen sind.

Das Verfahren ist mit dem Rabin-Verschlüsselungsverfahren verwandt.

Algorithmus

a. Wähle zufällig und stochastisch unabhängig zwei Primzahlen $p \neq q$, und berechne das Produkt $N = p \cdot q$.
   In der Praxis werden diese Primzahlen durch Raten einer Zahl und darauf folgendes Anwenden eines Primzahltests bestimmt.
b. Berechne $\varphi(N) = (p-1) \cdot (q-1)$, wobei φ für die Eulersche φ-Funktion steht.
c. Wähle eine Zahl *e*, für die gilt $1 < e < \varphi(N)$, die teilerfremd zu $\varphi(N)$ ist.

d. Berechne die Zahl *d* so, dass das Produkt $e \cdot d$ kongruent 1 bezüglich des Modulus $\varphi(N)$ ist, dass also $e \cdot d \equiv 1 \pmod{\varphi(N)}$ gilt.

Der **öffentliche Schlüssel** (**public key**) besteht dann aus
- *N*, dem *Primzahlprodukt* sowie
- *e*, dem *öffentlichen Exponenten*.

Der **private Schlüssel** (**private key**) besteht aus

- *d*, dem *privaten Exponenten* sowie
- *N*, welches allerdings bereits durch den öffentlichen Schlüssel bekannt ist.

p, *q* und $\varphi(N)$ werden nicht mehr benötigt und sollten nach der Schlüsselerstellung auf sichere Weise gelöscht werden.

Ein Zahlenbeispiel:

a. Für die beiden Primzahlen *p* und *q* wählt man *p* = 11 und *q* = 13. Damit wird *N* = 143.
b. Die eulersche φ-Funktion nimmt damit den Wert $\varphi(N) = \varphi(143) = (p-1)(q-1) = 120$ an.
c. Für die zu $\varphi(143) = 120$ teilerfremde neue Zahl *e* wählt man *e* = 23.
d. Mit dem erweiterten euklidischen Algorithmus berechnet man nun die Zahl *d* = 47 mit $e \cdot d \equiv 1 \pmod{\varphi(N)}$. Damit wird *d* = 47 der private Schlüssel, *e* = 23 und *N* = 143 der öffentliche Schlüssel.

Verschlüsseln von Nachrichten

Verschlüsselung

Um eine Nachricht *K* zu verschlüsseln, verwendet der Absender die Formel

$$C \equiv K^e \bmod N$$

und erhält so aus dem *Klartext* *K* den *Geheimtext* *C*.

Beispiel

Es soll die Zahl 7 verschlüsselt werden.
Der Nachrichtenabsender benutzt den veröffentlichten Schlüssel $N = 143$, $e = 23$ und rechnet

$$C \equiv 7^{23} \mod 143$$

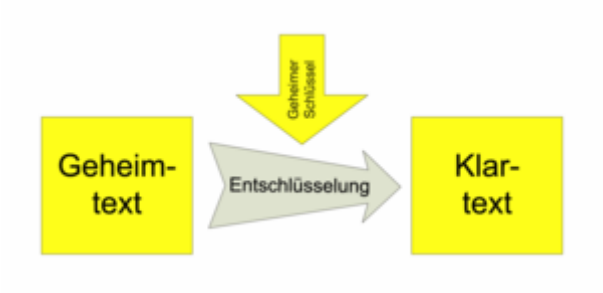
Zur Berechnung von $7^{23} \mod 143$ kann die Kongruenzarithmetik verwendet werden. Mit Hilfe der modularen Exponentiation berechnet man schnell:

$$7^{23} \mod 143 = (((7^2)^2 \cdot 7)^2 \cdot 7)^2 \cdot 7 \mod 143 = 2$$

Dabei wendet man nach jedem Rechenschritt auf die zu handhabenden Zahlen die Modulo-Operation (kurz: *mod*) an, um die Ergebnisse möglichst "klein" zu halten.

Aus dem Klartext 7 ist somit der Geheimtext 2 geworden.

Entschlüsseln von Nachrichten (Decodierung)



Entschlüsselung

Der Geheimtext *C* kann durch modulare Exponentiation wieder zum Klartext *K* entschlüsselt werden. Der Empfänger benutzt die Formel

$$K \equiv C^d \mod N$$

mit dem nur ihm bekannten Wert *d* sowie *N*.

Beispiel

$$K \equiv 2^{47} \mod 143 = ((((2^2)^2 \cdot 2)^2 \cdot 2)^2 \cdot 2)^2 \cdot 2 \mod 143 = 7$$

Aus $C = 2$ wird also wieder $K = 7$.

Signieren von Nachrichten

Um eine Nachricht *K* zu signieren, wird diese mit dem eigenen privaten Schlüssel verschlüsselt. Zum Prüfen der Signatur entschlüsselt der Empfänger die Nachricht mit dem öffentlichen Schlüssel des Senders und vergleicht diese mit dem empfangenen *K*. Wenn sie nicht übereinstimmen, ist die Signatur ungültig. Ist die Signatur gültig, dann kann sich der Empfänger sicher sein, dass derjenige, der das Dokument signiert hat, auch den privaten Schlüssel besitzt und dass niemand seit der Signierung das Dokument geändert hat. Es wird

also die Integrität und Authentizität garantiert, vorausgesetzt der private Schlüssel ist wirklich geheim geblieben.

Da asymmetrische Verfahren nicht geeignet sind um größere Datenmengen zu verschlüsseln, wird in der Praxis nicht die Nachricht selbst mit dem privaten Schlüssel verschlüsselt. Stattdessen wird der Hashwert H der Nachricht berechnet. Dieser bildet, meist zusammen mit einigen technisch notwendigen Informationen, wie z. B. das verwendete Hashverfahren, die Eingabe K^* der Verschlüsselung mit dem privaten Schlüssel. Diese liefert die eigentliche Signatur. Der Empfänger kann nun mit dem öffentlichen Schlüssel und der Signatur K^* bestimmen. Anschließend kann er selber den Hashwert der Nachricht bestimmen und mit dem in K^* gespeicherten vergleichen. Wenn die beiden Werte übereinstimmen, kann mit hoher Sicherheit davon ausgegangen werden, dass die Nachricht fehlerfrei übertragen wurde und nicht gefälscht ist.

Sicherheit

Public-Key-Verfahren wie RSA werden in der Praxis immer als hybride Verfahren in Verbindung mit symmetrischen Verfahren verwendet. Bei der Analyse der Sicherheit muss die Sicherheit des Public-Key-Verfahren und die praktische Sicherheit des Gesamtsystems betrachtet werden.

Sicherheit von RSA

Angenommen, der Angreifer kennt lediglich den öffentlichen Schlüssel, also die Werte N und e. Um den Geheimtext zu entschlüsseln, benötigt er zusätzlich d oder einen zu d modulo $\varphi(N)$ kongruenten Wert.

Wenn der Angreifer $\varphi(N)$ kennen würde, könnte er d leicht berechnen. Eine Möglichkeit dazu ist die Zerlegung von N in seine beiden Primfaktoren p und q. Diese Primfaktorzerlegung ist für große Zahlen mit den heute bekannten Verfahren praktisch nicht durchführbar. Es ist aber nicht bewiesen, dass es sich bei der Primfaktorzerlegung um ein prinzipiell schwieriges Problem handelt. Im Gegenteil, mit dem Quadratischen Sieb wurden bereits Zahlen mit über 100 Stellen faktorisiert. So gelang es z. B. Mathematikern der Universität Bonn 2005, im Rahmen eines Wettbewerbs eine einzelne von den RSA Laboratories vorgegebene 200-stellige Dezimalzahl zu faktorisieren. Die Faktorisierung begann Ende 2003 und dauerte bis Mai 2005. Unter anderem kam ein Rechnerverbund von 80 handelsüblichen Rechnern an der Universität Bonn zum Einsatz. Dies ist noch ein gutes Stück von den mindestens 300 Dezimalstellen heute üblicher Schlüssel entfernt. Im November 2005 zahlten RSA Laboratories für die Faktorisierung von RSA-640, einer Zahl mit 640 Bits bzw. 193 Dezimalstellen, eine Prämie von 20.000 US-$. Für die Faktorisierung von RSA-1024 (309 Dezimalstellen) oder gar RSA-2048 (617 Dezimalstellen) sind 100.000 $ bzw. 200.000 $ ausgelobt. Die wachsende Rechenleistung moderner Computer stellt für die Sicherheit von RSA kein Problem dar, zumal diese Entwicklung vorhersehbar ist: Der Nutzer kann bei der Erzeugung seines Schlüsselpaares darauf achten, dass die zu zerlegende Zahl groß genug ist, um während der Dauer der beabsichtigten Verwendung nicht berechnet werden zu können. Nicht vorhersehbare Entwicklungen wie z. B. die Fertigstellung eines leistungsfähigen Quantencomputers oder die Entdeckung eines wirklich neuen hocheffizienten Algorithmus sind dabei unerheblich. Beides ist kurzfristig gesehen sehr unwahrscheinlich.

Wir haben bereits gesehen, dass es nicht zwingend notwendig ist Faktorisierungsproblem zu lösen, um eine Nachricht zu entschlüsseln. Tatsächlich genügt es $\varphi(N)$ oder genauer ein Vielfaches des kgV von p-1 und q-1 zu kennen. Daraus lässt sich völlig analog wie bei der Schlüsselgenerierung ein passender Exponent d berechnen, da bei dieser Berechnung mittels des erweiterten Euklidischen Algorithmus die Primfaktorzerlegung nicht benötigt wird.

Angenommen zu einem N würden mehrere Paare von Exponenten (e,d) bzw. (e',d') verwendet. Dann könnte ein solches Vielfaches leicht aus irgend einem Paar (e',d') berechnet werden.

In diesem Spezialfall wäre die Entschlüsselung weit weniger schwierig als das Faktorisierungsproblem. Es ist offen, ob dies auch im allgemeinen Fall gilt. Das System von Rabin ist im Gegensatz zu RSA ein Public-Key-System, für das bewiesen ist, dass es mindestens so schwierig ist wie das Faktorisierungsproblem.

[Bearbeiten]

Praktische Sicherheit

RSA wird in der Regel in Hybridverfahren mit symmetrischen Verschlüsselungsverfahren kombiniert. Dabei wird zufällig ein Sitzungsschlüssel für eine symmetrische Verschlüsselung generiert, der dann per RSA verschlüsselt und zusammen mit der Nachricht übertragen wird.

Bei der Signatur wird nicht die gesamte Nachricht sondern nur ein Hashwert signiert. Der symmetrische Schlüssel bzw. der Hashwert sind dabei relativ kurz. Für den symmetrischen Schlüssel gilt also

$$K_{sym} < N = p \cdot q$$

Daher kann der symmetrische Schlüssel mit einer einzigen RSA-Verschlüsselung verschlüsselt werden. Für die Sicherheit von RSA sind Primzahlen mit über 100 Stellen erforderlich. Daher könnte ein symmetrischer Schlüssel mit über 200 Stellen verschlüsselt werden.

Als sehr sicher eingestufte Algorithmen zur symmetrischen Verschlüsselung sind 3DES und AES. Diese verwenden 112, 128 oder maximal 168 Bit oder etwa 40 Dezimalstellen. Damit ist eine Brute-Force-Angriff faktisch auszuschließen. Eine sichere Hashfunktion wie SHA-1 erzeugt Funktionswerte mit lediglich 160 Bit entsprechend etwas 50 Dezmalstellen bzw. 40 Hexadezimalziffern. Damit lassen sich Signaturverfahren mittels RSA realisieren, die nur einen Verschlüsselungsschritt benötigen.

Die Sicherheit und die Performance des Gesamtsystems wird durch die Sicherheit des Public-Key-Verfahrens limitiert, sofern nur als sicher eingestufte Algorithmen verwendet werden und die Wahl des Schlüssel als hinreichend zufällig betrachtet werden kann.

Vollständiges Beispiel

Vorarbeiten

Die oben genannten Schritte sollen nun an einem vollständigen Beispiel erläutert werden. Um einen Text zu verschlüsseln, müssen zunächst Buchstaben in Zahlen umgewandelt werden.

Dazu verwendet man in der Praxis z. B. den ASCII-Code. Hier sei willkürlich die folgende Zuordnung gewählt:

```
A=01 B=02 C=03 usw. (00 = Leerzeichen)
```

Darüber hinaus sei angenommen, dass jeweils drei Zeichen zu einer Zahl zusammengefasst werden. Die Buchstabenfolge AXT wird also zu 012420. Die kleinste zu verschlüsselnde Zahl ist dann 000000 (drei Leerzeichen), die größte 262626 (ZZZ). Der Modulus $N = p \cdot q$ muss also größer 262626 sein.

```
Klartext:  W I K  I P E  D I A
Kodierung: 230911 091605 040901
```

Schlüsselerzeugung

Zunächst werden geheim zwei Primzahlen gewählt, z.B. $p = 307$ und $q = 859$. Damit ergibt sich:

$$N = p \cdot q = 263713$$

$$\varphi(N) = (p-1) \cdot (q-1) = 262548$$

$e = 1721$ (zufällig, teilerfremd zu $\varphi(N)$)

$d = 1373$ (das multiplikative Inverse zu $e \bmod \varphi(N)$ mit Hilfe des Erweiterten euklidischen Algorithmus)

Öffentlicher Schlüssel: $e = 1721$ und $N = 263713$

Geheimer Schlüssel: $d = 1373$ und $N = 263713$

Verschlüsselung

$C_n = K_n^e \bmod N \quad$ für n=1,2,3(,...)
$C_1 = 230911^{1721} \bmod 263713$
$C_1 = 001715$
$C_2 = 091605^{1721} \bmod 263713$
$C_2 = 184304$
$C_3 = 040901^{1721} \bmod 263713$
$C_3 = 219983$

Entschlüsselung

$K_n = C_n^d \bmod N \quad$ für n=1,2,3(,...)
$K_1 = 001715^{1373} \bmod 263713$
$K_1 = 230911$
$K_2 = 184304^{1373} \bmod 263713$
$K_2 = 091605$
$K_3 = 219983^{1373} \bmod 263713$
$K_3 = 040901$

Signatur (Verschlüsselung mit dem geheimen Schlüssel)

$C_n = K_n^d \bmod N$
$C_1 = 230911^{1373} \bmod 263713$

$C_1 = 219611$
$C_2 = 091605^{1373} \bmod 263713$
$C_2 = 121243$
$C_3 = 040901^{1373} \bmod 263713$
$C_3 = 138570$

Verifikation (Entschlüsselung mit dem öffentlichen Schlüssel)

$K_n = C_n^e \bmod N$
$K_1 = 219611^{1721} \bmod 263713$
$K_1 = 230911$
$K_2 = 121243^{1721} \bmod 263713$
$K_2 = 091605$
$K_3 = 138570^{1721} \bmod 263713$
$K_3 = 040901$