Public-Key Verschlüsselung

Björn Thomsen

17. April 2006

Inhaltsverzeichnis

1 Einleitung

Whitfield Diffie und Martin E. Hellman haben in ihrem Bericht [1] im Jahre 1976 erstmals über Public-Key Verfahren (PKV) geredet. Es gibt Gerüchte, dass der amerikanische Geheimdienst schon vorher solche Verschlüsselungen kannte. Sie wollten der Kryptographie neue Impulse geben und ihr von einer rein mathematischen Seite begegnen. Sie sagten über Verschlüsselungen: „change this ancient art into a science“.

Ihre Motivation war der Einzug in das Computerzeitalter, in denen man über große Strecken miteinander kommuniziert. In diesem Fall ist es sehr schwer einen Schlüssel schnell und sicher zu dem Gesprächspartner zu bringen, um sicher über symmetrische Kryptographie Verfahren mit dem Partner zu kommunizieren. Das Verfahren was sie suchten sollte kein Verlangen nach einem sicheren Kanal haben. Ein Zuhörer darf mit der gegebenen Information, die er bekommt, nichts anfangen können. Dabei muss die Echtheit der Nachricht für jeden herausfindbar sein und gleichzeitig nicht imitiert werden können, selbst nicht vom Empfänger.

2 Wie funktioniert es

Das revolutionäre an PKV ist, dass es anstatt einem Schlüssel zum ver- und entschlüsseln zwei Schlüssel hat. Dadurch entfällt der oft schwere Schlüsselaustausch.

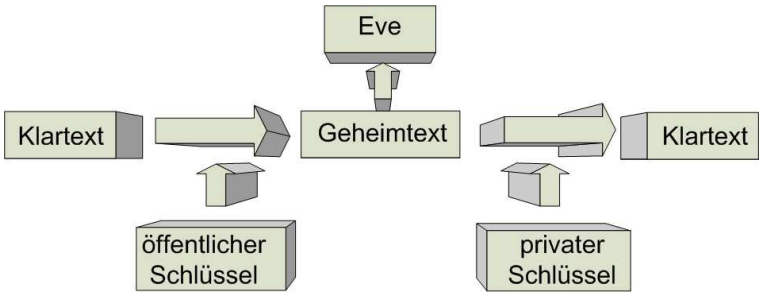

Abbildung 1: Weg der Nachricht bei PKV

Angenommen man hat ein Netz von Personen die miteinander sicher kommunizieren wollen. Jeder besitzt zwei Schlüssel, einen öffentlichen und einen privaten Schlüssel. Der öffentlicher wird zum ver- und der private Schlüssel zum entschlüsseln benutzt.

Der öffentliche Schlüssel wird über einen (auch unsicheren) Kanal zu dem Sender geschickt. Er kann auch an einer zentralen Stelle abgelegt werden (z.B. zentraler Server). Um der Forderung nachzukommen, dass ein Mithörer nichts mit den gegebenen Informationen anfangen kann, darf man aus dem öffentlichen Schlüssel nicht den privaten Schlüssel folgern dürfen. Den privaten Schlüssel muss man geheim halten, damit man nur selber damit die Nachricht entschlüsseln kann.

3 Vergleich mit symmetrischen Verfahren

Vorteile:

- Das Geheimnis bleibt klein. Jede Person muss nur seinen eigenen privaten Schlüssel geheim halten. Bei symmetrischen verfahren muss jeder alle Schlüssel geheim halten. Bei einem großen Netz von Personen sind das viele Schlüssel. Bekommt eine unberechtigte Person Zugang zu einer Schlüsselsammlung einer Person, sind alle Schlüssel wertlos und jeder muss mit jedem über sichere Kanäle (teilweise über große Strecken) die Schlüssel austauschen.
- Verminderung des Schlüsselverteilungsproblems. Die öffentlichen Schlüssel müssen nicht geheim gehalten werden. Eine Ablage an zentraler Stelle ist möglich.
- Leichtes einfügen einer neuen Person in die Gemeinde. Soll eine weitere Person mitspielen, bedeutet dies, einen erheblichen Mehraufwand bei symmetrischen Verschlüsselungen, ganz im Gegenteil zu Public-Key Verfahren. Bei symmetrischen Verfahren muss in dem Netzwerk jeder für jeden ein Schlüssel haben. Bei einem Neuzugang muss also jeder einen neuen Schlüssel erzeugen und mit ihm austauschen (über sichere Kanäle).

Nachteile:

- Langsam. Im Vergleich sind Public-Key-Verfahren extrem langsam in der Ver- und Entschlüsselung. ($\Rightarrow$ Hybridverfahren, z.B. PGP)
- Nicht sicheres mathematisches Fundament. Die meisten Public-Key-Verfahren beruhen auf schwierig zu berechnenden mathematischen Problemen. Es konnte bis jetzt noch kein Beweis geführt werden, dass ein Problem immer schwer zu lösen bleibt.
- Neues Verteilungsproblem. Die Echtheit eines öffentlichen Schlüssels muss Garantiert sein. Schafft es ein Angreifer seinen eigenen öffentlichen Schlüssel an die Stelle des orginalen Schlüssels zu setzen, kann er durch ein „Man-in-the-middle-Angriff“ die Nachricht lesen. Dies macht er indem er die Nachricht abfängt, mit seinem privaten Schlüssel entschlüsselt, liest und mit dem originalen öffentlichen Schlüssel verschlüsselt und weiter schickt. Man kann digitale Signaturen und/oder ein „Web of Trust“ [7]benutzen um dieses Problem zu umgehen.

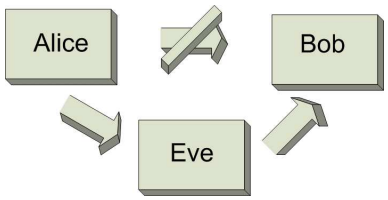

Abbildung 2: Man-in-the-middle-Angriff

4 Beispiel: RSA

Das Verfahren hat seinen Namen von seinen drei Erfindern Ron **R**ivest, Adi **S**hamir und Len **A**dleman. Es war das erste PKV und beruht auf der Schwierigkeit große Zahlen in seine Primfaktoren zu zerlegen.

4.1 Schlüsselerzeugung

Bevor man die Nachrichten übertragen will muss man erst einmal seinen privaten und öffentlichen Schlüssel erzeugen. Führe dafür folgende Schritte durch:

1. Wähle zwei (beliebige) Primzahlen p und q.
2. Berechne $n = pq$ und $\varphi(n) = (p-1)(q-1)$
   (φ ist die eulersche φ-Funktion. Sie gibt die Anzahl der teilerfremden Zahlen kleiner n an.)
3. Wähle eine natürliche Zahl e mit $1 < e < \varphi(n)$ und $ggT(e, \varphi(n)) = 1$
   (ggT ist der größte gemeinsame Teiler beider Zahlen)
4. Berechne eine natürliche Zahl d mit $1 < d < \varphi(n)$ und $de \equiv 1 \mod \varphi(n)$. Dieses kann man mittels den erweiterten euklidischen Algorithmus berechnen.
   (mod berechnet den Rest beim ganzzahligen Teilen Bspl.: $2 \equiv 9 \mod 7$, da $9 : 7 = 1$ Rest 2 ist)

Nun besteht der öffentliche Schlüssel aus n und e. Weiter ist der private Schlüssel d (und n).

4.2 Verschlüsselung

Jede natürliche Zahl m kleiner n kann verschlüsselt werden, in dem man den Schlüsseltext c berechnet mittels der Formel

$$c \equiv m^e \mod n$$

Wenn man dieses implementieren will, kann man mittels schneller Exponentation viel Rechenzeit sparen.

Bspl.: Berechnung von $3^{128} \mod 15$

$$9 \equiv 3^2 \mod 15$$
$$6 \equiv 9^2 \mod 15 \equiv 3^4 \mod 15$$
$$6 \equiv 6^2 \mod 15 \equiv 3^8 \mod 15$$
$$6 \equiv 3^{128} \mod 15$$

4.3 Entschlüsselung

Die Entschlüsselung funktioniert fast analog zur Verschlüsselung. Um m zu bekommen, führe folgendes durch

$$m \equiv c^d \mod n \equiv (m^e)^d \mod n$$

Das bei diesem Verfahren aber auch immer das „Richtige rauskommt“ sollte aber lieber bewiesen werden. Zu diesem Zweck benötigt man aber noch den

kleinen Satz von Fermat Ist p Primzahl, so gilt $a^{p-1} \equiv 1 \mod p$ für alle $a \in \mathbb{Z}$ mit p nicht teiler von a.

Beweis Wegen der Konstrukion des Verfahrens gilt $ed \equiv 1 \mod (p-1)(q-1)$. Also existiert eine ganze Zahl l, so dass gilt

$$ed = 1 + l(p-1)(q-1)$$

Nun kann man folgende Umformungen durchführen

$$(m^e)^d = m^{ed} = m^{1+l(p-1)(q-1)} = m(m^{l(p-1)(q-1)}) = m(m^{l(p-1)})^{q-1}$$

Aus dem kleinen Satz von Fermat folgt (wenn q nicht Teiler von m)

$$(m^e)^d \equiv m(m^{l(p-1)})^{q-1} \mod q \equiv m \mod q$$

Wenn q Teiler von m ist gilt $0 \equiv (m^{l(p-1)})^{q-1} \mod q$ (weil bei der Division durch q keine Rest bleibt) und somit auch

$$m^{ed} \equiv 0 \equiv m \mod q$$

Wir haben also festgestellt, dass q Teiler von $m^{ed} - m$ ist

Analog gilt das gleiche für p.

Insgesamt gilt, weil p und q verschieden sind, dass pq Teiler von $m^{ed} - m$ ist und somit

$$m^{ed} - m \equiv 0 \mod pq \Leftrightarrow m^{ed} \equiv m \mod pq \Leftrightarrow m^{ed} \equiv m \mod n$$

□

4.4 Blockchiffre

Es ist möglich mit RSA eine Art Blockchiffre zu realisieren. Man geht davon aus, dass das Alphabet genau N Zeichen hat und aus den Zahlen von 0 bis $N-1$ besteht. Besteht das Alphabet nicht aus diesen Zahlen, kann man eine Bijektion bestimmen, so dass dies erfüllt ist.

Die Länge der Blöcke k ergibt sich aus der Rechnung

$$k = \lfloor \log_N n \rfloor$$

wobei $n = pq$ gilt.

Nun berechnet man aus einem Block $m_1, m_2, \ldots, m_k$ die Zahl, die verschlüsselt wird durch die Berechnung

$$m = \sum_{i=1}^{k} m_i N^{k-i}$$

Als nächstes ergibt sich der Schlüsseltext c aus

$$c \equiv m^e \mod n$$

Jetzt kann es sein, dass c geschrieben zur Basis N die Länge $k+1$ hat. Grund dafür liegt im Abrunden bei der Berechnung von k. Deshalb ergibt sich für die Darstellung von c

$$c = \sum_{i=0}^{k} c_i N^{k-i}$$

Also bekommt man als Schlüsselblock $c = c_0 c_1 \ldots c_k$

4.5 Einmal mit Zahlen

Als Primzahlen wählt Bob $p = 11$ und $q = 13$. Dadurch ergibt sich $n = 143$ und $\varphi(11 \cdot 13) = 10 \cdot 12 = 2^3 \cdot 3 \cdot 5 = 120$. Demnach kann man $e = 7$ wählen.

d wird mittels dem euklidischen Algorithmus berechnet. Die Aufgabe lautet: Finde ein $d > 1$ mit $d \cdot 7 \equiv 1 \mod 120$.

$$120 = 17 \cdot 7 + 1 \Leftrightarrow 120 - 17 * 7 = 1$$

Da diese Rechnung in dem Restklassenring $\mod 120$ sich befindet, gilt

$$-17 \equiv 103 \mod 120.$$

Dadurch ergibt sich $d = 103$.

Alice und Bob haben sich auf das Alphabet $\Sigma = \{0, a, b, c\}$ geeinigt. Also erhält man eine Klartextblocklänge von $\lfloor \log_4 143 \rfloor = 3$. Weiter einigen sich beide, dass 0 für 0, 1 für a, 2 für b und 3 für c steht.

Alice will nun den Klartextblock *abb* an Bob schicken. Dies entspricht der Zahlenfolge 122 uns somit der Zahl

$$m = 1 \cdot 4^2 + 2 \cdot 4^1 + 2 \cdot 4^0 = 16 + 8 + 2 = 26$$

Diese wird zu c verschlüsselt mittels der Berechnung

$$c \equiv 26^7 \mod 143$$

Dieses Ergebnis kann man mittels schnelles Exponentieren bekommen:

$$104 \equiv 26^2 \mod 143$$

$$91 \equiv 104^2 \mod 143 \equiv 26^4 \mod 143$$

$$26 \equiv 104 \cdot 91 \mod 143 \equiv 26^6 \mod 143$$

$$104 \equiv 26 \cdot 26 \mod 143 \equiv 26^7 \mod 143$$

Also ist $c = 104$. Als Zahl zur Basis 4 dargestellt ergibt sich

$$c = 1 \cdot 4^3 + 2 \cdot 4^2 + 2 \cdot 4^1 + 0 \cdot 4^0$$

Dies entspricht dann dem Schlüsseltextblock *abb*0.

Bob erhält *abb*0 und wandelt es wieder zu $c = 104$ um. Nun berechnet er

$$104^{103} \mod 143$$

Hier kann man wieder schnelles Exponentieren anwenden und mit dem Taschenrechner nachrechnen:

$$91 \equiv 104^2 \mod 143$$

$$130 \equiv 91^2 \mod 143 \equiv 104^4 \mod 143$$

$$26 \equiv 130^2 \mod 143 \equiv 104^8 \mod 143$$

$$104 \equiv 26^2 \mod 143 \equiv 104^{16} \mod 143$$

$$91 \equiv 104^2 \mod 143 \equiv 104^{32} \mod 143$$

$$130 \equiv 91^2 \mod 143 \equiv 104^{64} \mod 143$$

$$104 \equiv 130 \cdot 91 \mod 143 \equiv 104^{96} \mod 143$$

$$78 \equiv 104 \cdot 130 \mod 143 \equiv 104^{100} \mod 143$$

$$91 \equiv 78 \cdot 91 \mod 143 \equiv 104^{102} \mod 143$$

$$26 \equiv 91 \cdot 104 \mod 143 \equiv 104^{103} \mod 143$$

Bob erhält also den Klartextblock 26 und kann dadurch die Nachricht *abb* ablesen.

4.6 Schlüssellänge

Die zur Zeit benutzten Schlüssel haben eine Stärke von 1024 bis 4098. Die Stärke bezieht sich auf die Länge von n in binärschreibweise. Wenn man es schafft die 2048 Zeichen starke Zahl „$RSA-2048$“[8] zu faktorisieren, erhält man von der Firma *RSA Security* 200.000$. Die letzte faktorisierte Zahl ist am 2.11.2005 faktorisiert worden. Es benötigte 30 2,2GHz-Rechner über 5 Monate um die 640 starke Zahl zu zerlegen. Dafür gab es auch ein Preisgeld von 20.000$.

5 Sicherheit

5.1 Semantische Sicherheit

Definition Ein Kryptosystem ist semantisch sicher, wenn es immun gegen Unterscheidbarkeit von Schlüsseltexten ist.

Anschaulich wünscht man sich ein digitales Gegenstück zu einem nicht durchsichtigen Briefumschlag. Sieht man die Briefe von außen, dann kann man nicht herausfinden was für ein Brief in dem Briefumschlag ist. Selbst dann nicht, wenn es nur 2 Möglichkeiten gibt.

als Spiel ausgedrückt:

1. Der Gute gibt dem Bösen seinen öffentlichen Schlüssel
2. Der Böse wählt zwei Nachrichten m_0 und m_1 und gibt sie dem Guten.
3. Der Gute wirft eine Münze. Bei Zahl verschlüsselt er m_0. Bei Kopf verschlüsselt er m_1. Das Ergebnis ist der Schlüsseltext c. Der Gute gibt dem Bösen den Schlüsseltext.
4. Der Böse rät, welcher Klartext verschlüsselt wurde. Er gewinnt, wenn er den richtigen Klartext rät.

Wenn der Böse nicht mit einer Wahrscheinlichkeit deutlich über 50% gewinnt ist das System semantisch sicher.

Die Frage ist nun natürlich, ob RSA sicher ist. Die Antwort: NEIN. Folgendes Szenario

- Der Böse generiert m_0 und m_1.
- Der Gute wählt m_0 oder m_1 aus.
- der Gute sendet die verschlüsselte Nachricht an den Bösen.
- Der Böse verschlüsselt selber beide Nachrichten (öffentlicher Schlüssel bekannt) und vergleicht sie mit der von dem Guten erhaltenen Nachricht.

Satz Jedes deterministische PKV ist nicht semantisch sicher.

Diese Aussage ist das Resultat davon, dass deterministische PKV bei gleichem zu verschlüsselnden Text immer den gleichen verschlüsselten Text als Ergebnis haben. Deshalb ist das obere Szenario auch auf jedes deterministische PKV erweiterbar.

Es gibt Abwandelungen von RSA die semantisch sicher sind.

5.2 Adaptive-Chosen-Ciphertext-Sicherheit

Semantische Sicherheit bietet Schutz gegen passive Angriffe. Bei aktiven Angriffen muss man ein bisschen mehr fordern. Dazu gibt es natürlich wieder ein Spiel:

1. Der Gute gibt dem Bösen seinen öffentlichen Schlüssel
2. Der Böse kann sich vom Guten Chiffretexte seiner Wahl entschlüsseln lassen. Damit kann er versuchen, die Auswahl im nächsten Schritt vorzubereiten.
3. Der Böse wählt zwei Nachrichten m_0 und m_1 und gibt sie dem Guten.
4. Der Gute wirft eine Münze. Bei Zahl verschlüsselt er m_0. Bei Kopf verschlüsselt er m_1. Das Ergebnis ist der Schlüsseltext c. der Gute gibt dem Bösen den Schlüsseltext.
5. Der Böse kann sich vom Guten Chiffretext seiner Wahl entschlüsseln lassen, nur nicht c. Damit kann er versuchen, die Auswahl im nächsten Schritt vorzubereiten.
6. Der Böse rät, welcher Klartext verschlüsselt wurde. Er gewinnt, wenn er den richtigen Klartext rät.

Wenn der Böse nicht mit einer Wahrscheinlichkeit deutlich über 50% gewinnt ist das System gegen Adaptive-Choosen-Ciphertext-Angriffe sicher. Ist dies der Fall, dann kann der Böse garnichts über den Klartext aus dem Chiffretext herausziehen.

5.3 Sicherheitsbeweise

Da es in der Mathematik im Moment keine nachweisbar schwierigen Probleme gibt, können die Beweise auch nur bedingt durchgeführt werden. Zu diesem Zweck nimmt man an, dass bestimmte Berechnungen schwer sind und unter diesem Umstand dann auch sicher sind.

6 Anwendung

PKV werden heutzutage häufig zum Schlüsselaustausch eines symmetrischen Schlüssels und zur Sicherung der Integrität benutzt. Bekannte Beispiele sind im E-Mail Verkehr (Open-)PGP und S/MIME. Weiter werden sie auch bei kryptographischen Protokollen bei SSH verwendet oder bei digitalen Signaturen/Unterschriften.

Literatur

[1] Diffie, Whitfield und Hellman, Martin E. : New Directions in Cryptography; http://www-ee.stanford.edu/ hellman/publications/24.pdf; 1976

[2] Buchmann, Johannes: Einführung in die Kryptographie; Kapitel 9; Springer; 2003

[3] Schulz, Ralph-Hardo: Codierungstheorie; Kapitel 4; Vieweg; 2003

[4] http://www.cs.uni-potsdam.de/ti/lehre/04-Kryptographie/slides/rsa-attacken.pdf

[5] http://de.wikipedia.org/wiki/RSA-Kryptosystem

[6] http://de.wikipedia.org/wiki/Asymmetrisches_Kryptosystem

[7] http://de.wikipedia.org/wiki/Web_of_trust

[8] RSA Challenge http://www.rsasecurity.com/rsalabs/node.asp?id=2093#RSA2048