Verschlüsselung am Beispiel PGP

Thijs Metsch, Martin Kolleck, Michael Krimgen

24. November 2003

1 Was ist PGP

PGP steht für Pretty Good Privacy und ist ein Verschlüsselungsprogramm, das hauptsächlich für Emails, aber auch alle anderen elektronischen Dateien gedacht ist. Außerdem kann es zur digitalen Signatur verwendet werden. PGP wurde 1984 von Philip Zimmermann [2], einem Software Ingenieur, der sich auf dem Gebiet der Kryptographie spezialisiert hatte, entwickelt. Die Idee heirzu kam vor allem duch den Hintergrund des kalten Krieges, da zur dieser Zeit die Angst vor Spionage und einem Atomkrieg auf Grund der fortschreitenden Entwicklung von elektronischen Kommunikationswegen immer größer. Als 1991 PGP für den Privatgebrauch als Freeware veröffentlicht wurde, begann ein endloser Streit um diese Entwicklung der Kryptographie für die breite Masse. Polizei, Militär und Geheimdienst befürchteten, dass damit kriminellen Aktivitäten Tür und Tor geöffnet werde, weil diese ohne abgehört zu werden miteinander kommunizieren könnten. Zimmermann musste sogar eine Freiheitsstrafe verbüßen. Gleichzeitig wurde er aber vom Militär zu Schulungen engagiert.

2 PGP Aktuell

Die aktuelle Version PGP 8.0.3 ist für den privaten Gebrauch und für gemeinnützige Organisationen kostenlos unter [1] erhältlich. Eine Anleitung zur Funktionsweise ist unter [3] und [4] zu finden. Als gedruckte Version empfiehlt sich [5]. Die uneingeschränkte Version für kommerzielle Zwecke kostet 260 Euro.

3 Wie funktioniert PGP

3.1 Verschlüsselung allgemein

Grundsätzlich gibt es zwei Verschlüsselungsmethoden: die konventionelle Verschlüsselung, bei der Sender und Empfänger den gleichen, geheimen Schlüssel verwenden und die Public Key Verfahren, bei denen jeder Teilnehmer ein Schlüsselpaar bestehend aus privatem und öffentlichem Schlüssel hat.
Als konventionelles Verfahren benützt PGP den symmetrischen IDEA (international data encryption standard). Dieser Algorithmus wurde von von Lai und Massey vom ETH Zürich entwickelt und ist patentrechtlich geschützt (Ascom Systec AG, Schweiz). Der Klartext wird in 64 Bit Blöcke aufgeteilt und mit einem 128 Bit Schlüssel verschlüsselt. Dies geschieht einfach, indem verschiedene mathematische Operationen wie XOR und modulo auf die Blöcke angewandt werden. Die Entschlüsselung erfolgt mit dem gleichen Schlüssel. Daher die Bezeichung als symmetrisches Verfahren.
Als asymmetrisches Verfahren benützt PGP den RSA Algorithmus. Dieser wurde 1978 von Rivest, Shamir und Adlemann entwickelt. Vorrausetzung hierfür ist der Euler-Fermat Satz:

$$x^{\varphi(n)} \equiv 1 (mod\, n) \text{ für ggt(x, n)=1}$$

hierbei ist $\varphi(n)$ die $Eulersche\ \varphi - Funktion$. Bei dieser gilt im Falle $n = p \cdot q$ für zwei Primzahlen p und q: $\varphi(n) = (p-1)(q-1)$

Schlüsselerzeugung:

- Wahl zweier (sehr großer) Primzahlen p und q
- Berechung der öffentlichen Schlüssels n = pq und
- Wahl des 2. öffentlichen Schlüssels e mit $1 < e < n$
- Wahl von d mit $ed \equiv 1 (mod \varphi(n)) \rightarrow ed = 1 + k \cdot \varphi(n)$

Die beiden öffentlichen Schlüssel werden dem Sender bekannt gegeben. Der geheime Schlüssel darf nur dem Empfänger bekannt sein und p, q und $\varphi(n)$ sollten zur Sicherheit gelöscht werden, da sie nicht weiter gebraucht werden.
Verschlüsselung:
Zur Verschlüsselung der Nachricht M berechnet der Sender

$$G = M^e\ mod\, n$$

und schickt G an den Empfänger. Die Nachricht M muss hierbei (wegen Eindeutigkeit) kleiner als n sein.

Entschlüsselung:
Der Empfänger entschlüsselt die Nachricht G, indem er sie mit seinem geheimen Schlüssel d potenziert und mod n rechnet.

$$\begin{aligned} G^d &\equiv (M^e)^d \ (mod\, n) \\ &\equiv M^{e \cdot d} \ (mod\, n) \\ &\equiv M^{1+k\varphi(n)} \ (mod\, n) \\ &\equiv M^1 \cdot M^{k\varphi(n)} \ (mod\, n) \\ &\equiv M^1 \equiv M \ (mod\, n) \end{aligned}$$

Anmerkung: Der Euler-Fermat Satz gilt in Zeile 4 auch wenn $ggT(m,\, n) \neq 1$, dies läßt sich durch Umformung zeigen.

Ein möglicher Mithörer kennt zwar n, e und M, kann die Nachricht aber nicht entschlüsseln, da er p und q bzw. $\varphi(n)$ nicht kennt. Die Verschlüsseling beruht hierbei auf der Schwierigkeit n zu faktorisieren bzw. $\varphi(n)$ zu bestimmen. Bei einer Bitlänge von 1024 Bit würde ein normaler Pentium PC bei den besten zur Zeit bekannten Algorithmen ca. 1000000000 Jahre brauchen, um eine Zahl solcher Länge zu faktorisieren.

3.2 Verschlüsselung von Nachrichten mit PGP

Zur Verschlüsselung von Nachrichten verwendet PGP ein Hybridverfahren aus einem symmetrischen und asymmetirschen Verfahren. Zunächst wird der Klartext mit dem IDEA Verfahren verschlüsselt. Der geheime Schlüssel herizu wird dann mit dem RSA Verfahren verschlüsselt. Der Grund hierfür liegt darin, dass die Verschlüsselung des ganzen Textes mit RSA zuviel Rechenzeit in Anspruch nehmen würde.

3.3 Digitale Signatur mit PGP

Die digitale Signatur oder elektronische Unterschrift soll den Verfasser einer Nachricht authentifizieren. Hierzu wird zunächst eine Einweg Hashfunktion (MD5 oder SHA1) auf den Text angewand, wodurch man einen eindeutigen fingerprint oder Fingerabdruck der Nachricht erhält. Diese Prüfsumme verschlüsselt man mit seinem privaten Schlüssel und verschickt das ganze. Der Empfänger errechnet ebenfalls die Prüfsumme und vergleicht diese mit der erhaltenen, die er mit dem öffentlichen Schlüssel des Senders entschlüsselt hat. Sind beide Prüfsummen identisch, so kann sich der Empfänger sicher

sein, dass die Nachricht vom Besitzer des öffentlichen Schlüssels stammt und nicht verändert wurde.

Literatur

[1] http://www.pgp.com

[2] http://www.philzimmermann.com/index.shtml

[3] http://www.foebud.org/pgp/

[4] http://www.helmbold.de/pgp/pgp6.5/wasistpgp.htm

[5] P. Zimmernmann, PGP, Verlag ART D'AMEUBLEMENT, Bielefeld, 1999, ISBN 3-9802182-9-5