

Verschlüsselte Datenübertragung per E-Mail



Abb. 1: Verschlüsselungsmaschine Enigma (Ausschnitt)

Das Recht auf Privatsphäre steht eigentlich jedem zu. Nicht umsonst gibt es schließlich das Brief- oder das Fernmeldegeheimnis, deren Verletzung wir alle als Eingriff in persönliche Angelegenheiten empfinden.

Um so erstaunlicher ist es, wie sorglos viele Leute persönlichste Dinge per E-Mail schreiben – ohne sich darüber im Klaren zu sein, dass eine E-Mail nichts anderes ist als eine völlig offene Textdatei, die auf allen Systemen, die für den Transport zuständig sind, nicht nur gelesen, sondern auch problemlos automatisch kopiert und archiviert werden kann; eine elektronische Postkarte sozusagen.

Das „Abhören“ des E-Mail-Verkehrs lässt sich nicht vermeiden, aber die Daten können so übermittelt werden, dass sie durch Dritte nicht gelesen werden können. Dazu werden die Daten vor dem Versand verschlüsselt und vom Empfänger – mit dem passenden Schlüssel – wieder entschlüsselt.

Public Key Verschlüsselung

Klassische Methoden zur Verschlüsselung benutzen nur einen Schlüssel. Der Sender verschlüsselt seine Nachricht mit diesem Schlüssel, und der Empfänger entschlüsselt ihn mit demselben wieder. Solche Verfahren heißen *symmetrische Verschlüsselung* (Abb. 2).

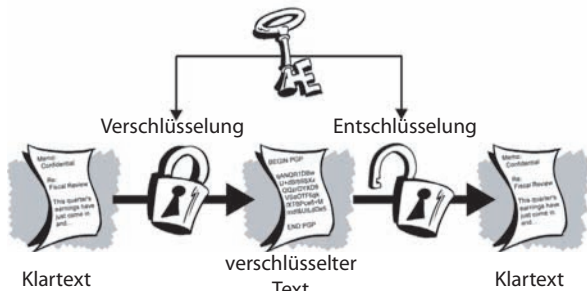


Abb. 2: Symmetrische Verschlüsselung

Damit das funktioniert, muss der Empfänger vorher den Schlüssel bekommen haben, und zwar auf einem sicheren Kommunikationskanal, da sonst Unbefugte in Kenntnis des Schlüssels gelangen könnten. Wenn man aber bereits über einen sicheren Kommunikationskanal verfügt, braucht man auch nicht mehr zu verschlüsseln (wenn man von Anwendungen wie einem Codebuch für den Funkverkehr und Ähnlichem absieht).

Public Key Verschlüsselung (auch: *asymmetrisches Verschlüsselung*) beseitigen dieses Problem, indem zwei Schlüssel erzeugt werden: Der Öffentliche, der über beliebige Kommunikationskanäle verschickt werden kann und der private, den nur der Besitzer kennt. Idealerweise ist der private Schlüssel nicht mit dem öffentlichen rekonstruierbar. Der Sender verschlüsselt die Nachricht mit dem öffentlichen Schlüssel des Empfängers. Entschlüsselt wird die Nachricht dann mit dem privaten Schlüssel des Empfängers (Abb. 3).

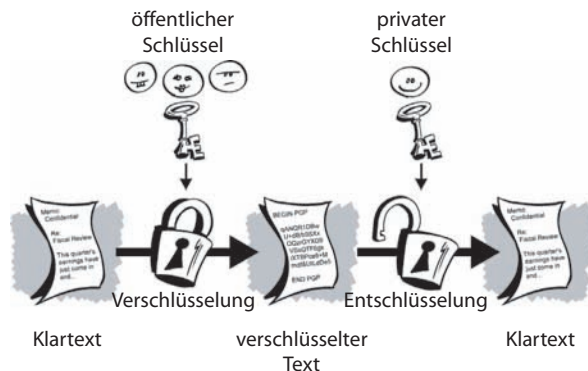


Abb. 3: Asymmetrische Verschlüsselung

Nach diesem Schema kann man demnach effektiv verschlüsseln, ohne über einen sicheren Kommunikationskanal zu verfügen. Dieses Verfahren wird zum Beispiel in der Verschlüsselungssoftware *PGP* („*Pretty Good Privacy*“) und *GnuPG* („*GNU Privacy Guard*“) angewandt.

Ein ganz wichtiger Punkt ist aber die Geheimhaltung des privaten Schlüssels. Er darf auf keinen Fall in fremde Hände geraten, auch nicht über das Netz verbreitet werden.

Digitale Unterschriften (Signaturen)

Digitale Unterschriften sollen die Authentizität einer Nachricht beweisen. Würden Nachrichten von offizieller Seite signiert, wäre es deutlich schwerer, mit gefälschten Nachrichten Unruhe oder Schaden anzurichten (Echtes Beispiel: Ein Trojaner, verschickt als Patch eines bekannten Webrowsers).

Eine digitale Signatur wird mit Hilfe des privaten Schlüssels aus dem Text erzeugt. Diese kann dann vom Empfänger mit dem öffentlichen Schlüssel des Senders überprüft werden (Abb. 4).

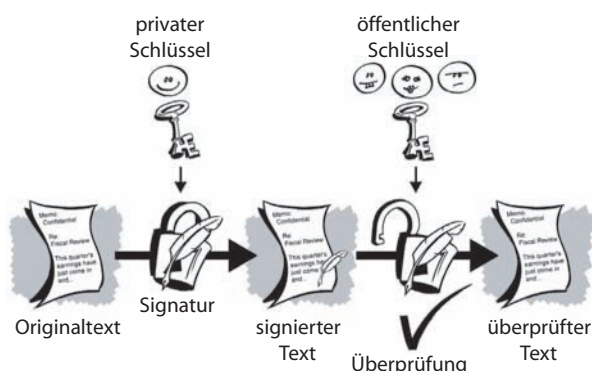


Abb. 4: Signatur

Dabei wird nicht nur der Absender (nur der kennt den privaten Schlüssel) überprüft, sondern auch, ob der Text unverändert angekommen ist.

Web of Trust

Eine Schwachstelle der Public Key Algorithmen ist die Verbreitung der öffentlichen Schlüssel. Ein Benutzer könnte einen öffentlichen Schlüssel mit falscher User ID in Umlauf bringen. Wenn dann mit diesem Schlüssel Nachrichten kodiert werden, kann der Eindringling die Nachrichten dekodieren und lesen. Wenn er sie dann noch mit einem echten öffentlichen Schlüssel kodiert an den eigentlichen Empfänger weiterleitet, fällt dieser Angriff nicht einmal auf. In der Literatur heissen solche Angriffe *man-in-the-middle attacks*, sie stellen auch bei vielen anderen Protokollen eine Bedrohung dar.

Die von PGP (und damit auch von GnuPG) gewählte Lösung besteht im Unterschreiben von Schlüsseln. Ein öffentlicher Schlüssel kann von anderen Leuten unterschrieben werden. Diese Unterschrift bestätigt, dass der Schlüssel zu der in der UID angegebenen Person gehört. Der Benutzer kann festlegen, welchen Unterschriften er wie weit traut. Ein Schlüssel gilt als vertrauenswürdig, wenn er von Leuten unterzeichnet wurde, denen man vertraut. Wenn man Schlüssel unterzeichnet, sollte man sich sicher sein, dass man die Identität desjenigen, dessen Schlüssel man unterschreibt, genau kennt. Eine Möglichkeit ist es, den Schlüssel persönlich bekommen zu haben, eine andere, den Fingerprint (eine aus dem Schlüssel gebildete Quersumme) über zuverlässige Kanäle zu vergleichen.

Grenzen der Sicherheit

Wenn man Daten vertraulich halten will, sollte man sich nicht nur Gedanken über die Sicherheit des Verschlüsselungsalgorithmus machen, sondern über die Systemsicherheit allgemein. Die in GnuPG verwendeten Algorithmen gelten gemeinhin als nicht zu knacken. Daraus zu schliessen, dass alle verschlüsselten Daten sicher seien, ist naiv.

Es gibt auch noch andere Formen von Angriffen. Anfang Februar 1999 tauchte zum Beispiel ein Word Trojaner auf, der private PGP Schlüsselbunde auf der Festplatte suchte und via ftp verschickte (Meldung im Heise Newsticker vom 03.02.99). Ein privater Schlüsselbund lässt sich, insbesondere bei schlechtem Passwort, deutlich leichter knacken als eine einzelne Datei.

Denkbar sind auch Trojaner, die Tastatureingaben weiterleiten. Auf *Slashdot* wurde im November 2001 berichtet, dass das FBI tatsächlich versucht, mittels bekannter Schwachstellen Backdoors zu installieren, um Verschlüsselung zu umgehen. Falls man die Nachrichten entschlüsselt auf dem Rechner lagert, können sie dort natürlich auch gelesen werden. Aufwändiger, aber technisch möglich ist es, die Abstrahlung des Monitors zu messen und sichtbar zu machen, so dass der Bildschirminhalt mitgelesen werden kann. Dann nützt es auch nichts, eine verschlüsselte Datei nur zum Lesen zu entschlüsseln. Wiederum bei *Slashdot* konnte man im Sommer 2001 lesen, dass das FBI im Rahmen einer Ermittlung gegen Mafiosi Wanzen in den Tastaturen angebracht hat, um deren PGP Key zu erfahren.

Die obigen Möglichkeiten sollen keine Paranoia hervorrufen, sondern nur darauf hinweisen, dass Verschlüsselung von Daten nur ein Bau-stein eines Sicherheitskonzeptes sein kann.

Mac-Software für die Verschlüsselung

Für die Verschlüsselung von E-Mails braucht es einerseits eine Software für die Schlüsselerzeugung und -verwaltung (sofern man nicht über gute Unix-Kenntnisse verfügt), andererseits ein Plug-In für das verwendete E-Mail-Programm, damit dieses E-Mails überhaupt ver- und entschlüsseln kann. Bei der hier vorgestellten Software handelt es sich durchwegs um OpenSource- bzw. Freeware-Lizenzen.

Erzeugung und Verwaltung von Schlüsseln:

GnuPG, <http://www.gnupg.org/>

Plug-Ins für E-Mail-Clients:

Microsoft Entourage:

EntourageGP, <http://entouragepgp.sourceforge.net/>

Apple Mail:

GPGMail, <http://www.sente.ch/software/GPGMail/English.lproj/GPGMail.html>

Eudora:

integriert

Netscape, Mozilla, Firebird:

Enigmail, <http://enigmail.mozdev.org/>

aoe, Nov. 04

Beratung
Service und Support
Hard- und Software
www.ingeno.ch, info@ingeno.ch
www.nettocomputer.ch

Ingeno Computer AG
Netto Computer AG
Netto Computer AG
Netto Computer AG
Ingeno Data AG

8047 Zürich, Fellenberberstrasse 291, 044 406 12 12
8047 Zürich, Fellenberberstrasse 291, 044 406 12 12
8305 Dietlikon, Brandbachstrasse 8, 044 805 75 05
8200 Schaffhausen, Grabenstrasse 2, 052 620 49 55
4001 Basel, Güterstrasse 2, 061 366 11 11