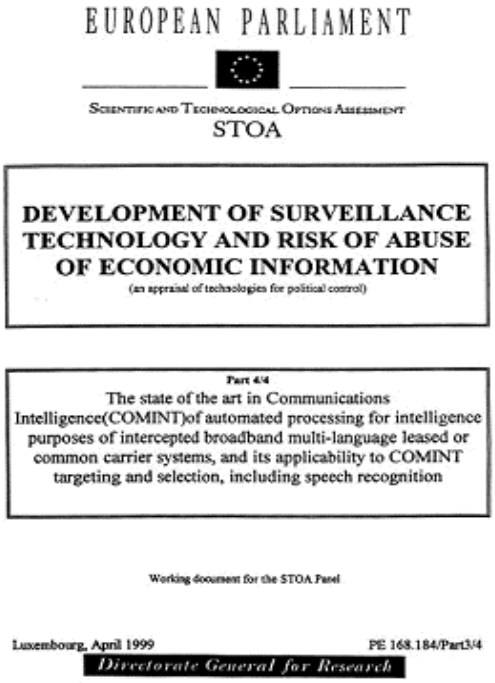
von Kai Raven

Version 3.6



EUROPEAN PARLIAMENT

SCIENTIFIC AND TECHNOLOGICAL OPTIONS ASSESSMENT
STOA

DEVELOPMENT OF SURVEILLANCE TECHNOLOGY AND RISK OF ABUSE OF ECONOMIC INFORMATION
(an appraisal of technologies for political control)

Part 4/4
The state of the art in Communications Intelligence(COMINT)of automated processing for intelligence purposes of intercepted broadband multi-language leased or common carrier systems, and its applicability to COMINT targeting and selection, including speech recognition

Working document for the STOA Panel

Luxembourg, April 1999 PE 168.184/Part3/4
Directorate General for Research

"If privacy is outlawed, only outlaws will have privacy."
Phil Zimmermann

"PGP for privacy and authentication in the nets!"
Kai Raven

Vorbemerkung oder
"Was soll das Ganze ?"

Ich habe 1995 damit angefangen, mich mit PGP zu beschäftigen, da ich PGP als eines der wichtigsten Werkzeuge zur Kommunikation über das Internet ansehe.
Nach meiner Überzeugung stehen wir mit dem Beginn der allseitigen globalen Vernetzung am Anfang vom Ende jeder schriftlichen Kommunikation auf herkömmliche Art: Brief, Fax und Postkarte.
Damit werden krypto- und steganografischen Algorithmen und ihre Umsetzung in bedienbare Programme wie PGP ein hoher Stellenwert zukommen, denn sie sind es, die das Recht von Privatpersonen wie Firmen auf private Kommunikation sichern werden.
PGP garantiert mir und allen anderen Netizens, unter dem Aspekt des Datenschutzes und der Freiheit des Internets, sichere und freie Kommunikation mit anderen Menschen, die vielleicht stärker ist als die herkömmliche Verständigung über Telefon oder Brief.
Mittlerweile schreiben wir das Jahr 1998 und im Laufe der Zeit sind neue Enthüllungen und Erkenntnisse zum Ausmass der Überwachung, Spionage und Kontrolle, die sowohl Unternehmen wie Individuen gleichermassen betrefen, bekanntgeworden.
ECHELON, NEWSWATCH, WEBBLOCK, DER GROSSE LAUSCHANGRIFF, DAS WASSENAAR ABKOMMEN, DIE TELEKOMMUNIKATIONSÜBERWACHUNGSGESETZE und ENFOPOL sprechen eine allzu deutliche Sprache und zeigen uns umso deutlicher, wie wichtig frei verfügbare und benutzbare Kryptografieprogramme und das Recht auf informationelle Selbstbestimmung geworden sind.
Damals habe ich mich durch das PGP-Handbuch der FoebuD gewühlt, der ich an dieser Stelle meinen Dank aussprechen möchte, neben den Schreibern in den PGP Newsgroups, die mir bei mancher Nachfrage halfen.
Ebenso mein Dank an Alexander Svensson, Johannes Posel, Michael Uplawski, Heiko Friedrich, Lutz Donnerhacke, Bernard Azzi und Stefan Kelm für erhaltene Tips.
Es gab noch kein PGP 5, sondern das DOS-Programm PGP 2.6.X und in der ersten Zeit habe ich mit Hilfe des Handbuches die Möglichkeiten von PGP per Kommandoprompt erkundet.
Danach setzte ich auch PGP-Shells ein, die die Kommandos in Buttons versteckten. Jetzt gibt es die PGP 5/6 Versionen und ich denke, die Mehrheit der PGP-Benutzer wird langfristig PGP 5/6 oder andere grafische Versionen von PGP einsetzen.
Viele neue PGP-User werden die Kommandozeile nicht mehr kennenlernen und damit auch nicht die Möglichkeit, PGP "from the roots" zu entdecken. Trotzdem würde ich jedem Anfänger raten, auch mit PGP 2.6.X zu arbeiten.
Um die Hilfe, die ich erhalten habe, weiterzugeben und zum erfolgreichen Arbeiten mit PGP 5/6 beizutragen, habe ich diese Anleitung geschrieben.
So, dass war's von mir und jetzt viel Spass mit der Anleitung und mit PGP !

Kleines Lexikon und Abkürzungen oder
"Was ist mit XY gemeint ?"

1xMKrT	1 x Mausklick mit rechter Maustaste
1xMKlT	1 x Mausklick mit linker Maustaste
Decryption	Entschlüsselung eines Textes oder einer Datei
Default Key	Der Schlüssel, mit dem man standardmäßig signieren will
E-Mail	Nachricht an eine Person oder Mailingliste
E-Mail Signatur	ein vierzeiliger Textblock am Ende einer E-Mail, die zur Information persönliche Angaben des Autoren enthält
Encryption	Verschlüsselung eines Textes oder einer Datei
Export	ein Public Key wird aus dem Pubring herauskopiert und als Datei abgelegt
File	Text- oder Binärdatei
Fingerprint	"Fingerabdruck" eines Keys. Eine Zahlenreihe, die bei der Schlüsselerzeugung als MD5/SHA-1 Prüfsumme der Schlüsselbits erstellt wird
Hash(wert)	zu einem Ursprungstext von beliebiger Länge wird über eine mathematische Transformation, der Hash Funktion, eine Zeichenkette mit feststehender Länge berechnet und komprimiert. Auch *Message Digest* oder *Textprüfsumme* genannt.
Key-ID	Schlüsselnummer. Eine Ziffer mit der Syntax "0xXXXXXXXX", die es für jeden Key nur einmal gibt (ähnlich den Nummern, die man auf normalen Sicherheitsschlüsseln findet)
Passphrase	Password Phrase. Ein langes Passwort, auch »Mantra« genannt, das über die Bildung des Secret Keys der Verschlüsselung des Private Keys dient
Posting	Nachricht in eine Newsgroup des Usenets
Public Key	öffentlicher Schlüssel mit dem der Sitzungsschlüssel verschlüsselt wird. Enthalten in der Datei »pubring.pkr«
PGPdisk	PGP Anwendung mit der CAST-128 verschlüsselte Containerdateien verwendet werden, die als zusätzliches Laufwerk eingebunden werden

PGPkeys	PGP Anwendung zum Schlüsselmanagement (Erzeugung, Import, Export, Revocation, Suche)
PGP Signatur	auch "Digitale Signatur". Ein PGP Block am Ende einer E-Mail, der die Textprüfsumme des Textes darstellt (ähnlich der CRC-Prüfsumme bei ZIP Archiven) und die Überprüfung der Authentizität eines Textes ermöglicht
PGPtools	PGP Anwendung, die alle PGP Funktionen als separate Buttonleiste zusammenfasst
PGP Zertifikat	Eine Beglaubigung durch eine Zertifizierungsinstanz (CA), die die Zugehörigkeit des Schlüssels und seiner Signatur zu einer bestimmten Person bescheinigt
Pubring	Public Ring/Public Key Ring. Öffentlicher Schlüsselbund, die Datei »pubring.pkr«
Private Key	privater Schlüssel mit dem der Sitzungsschlüssel entschlüsselt wird. Enthalten in der Datei »secring.skr«
Revocation	Rückzugsurkunde. Ein Public Key, der mit dem Merkmal "ungültig/zurückgezogen" veröffentlicht wird
SDA (Self decrypting archive	Eine mit CAST-128 verschlüsselte Archivdatei, die sich nach Passphraseeingabe selbst entpackt
Secret Key	Hashwert, gebildet aus der Passphrase zur Verschlüsselung des Private Keys
Secring	Secret Ring/Secret Key Ring. Privater (geheimer) Schlüsselbund, die Datei »secring.skr«
Session Key	zufälliger Schlüssel, mit dem der Klartext konventionell mittels IDEA verschlüsselt wird. Der Session Key wird mit dem Public Key des Empfängers verschlüsselt
Trust	Vertrauensparameter. Vertrauen, dass man einem User zubilligt und damit auch das Vertrauen in einen Key, der von diesem User signiert wurde
User-ID	Besitzerkennung. Der Name (+ der E-Mailadresse) des Schlüsselbesitzers
Validity	Gültigkeit, Authentizität eines Public Keys. Bezeichnet den Grad, inwieweit der Public Key wirklich dem User zuzuordnen ist
Verification/verify	Überprüfung der PGP-Signatur

Überblick oder
"Was ist PGP ?"

Diese Anleitung setzt den Einsatz von **PGP Versionen ab Versionsnummer 6.0** voraus, die RSA-Keys erzeugen können, wie z. B. bei der PGP Version 6.0 for Business Security oder der internationalen Version PGP 6.0.2i, da ein Grossteil der PGP Anwender noch PGP 2.6.3 einsetzt, das nicht in der Lage ist DH/DSS Keys zu verarbeiten.

Einige PGP Versionen der 5er und 6er Reihe können gar nicht oder nur eingeschränkt RSA Keys erzeugen, bzw. verwenden (siehe dazu die Tabelle PGP-Versionen).

Ein weiterer Vorteil der internationalen Version liegt im Vorliegen des kompletten Sourcecodes.

PGP 5.0 als grafisches Benutzerprogramm war die neueste PGP-Entwicklung seit Erscheinen der PGP Version 2.6.3.

Die Version 2.6.3, die kommandozeilenorientiert arbeitet und über die Eingabeaufforderung oder über grafische Shells (Benutzeroberflächen) bedient wird, erfreut sich weiterhin (teilweise aus guten Gründen) grosser Beliebtheit. Es gibt einige Unterschiede zwischen beiden Versionen, die auch in dieser Anleitung behandelt werden und Einschränkungen im Funktionsumfang, der bei der Version 6 nicht so gross ist wie bei der Version 2.6.3. Wer sich näher mit der Version 2.6.3 beschäftigen möchte, kann sich die informative Comp.Security.PGP.FAQ, mit der deutschen Übersetzung von Michael Uplawski der comp.security.pgp FAQ von A. "Galactus" Engelfriet herunterladen.

PGP ist die Abkürzung für "Pretty Good Privacy", was sich etwas holprig mit "Ganz guter Geheimhaltung" übersetzen lässt. PGP wurde von dem amerikanischen Programmierer Phil Zimmermann geschrieben und im Jahr 1991 als Freeware veröffentlicht. Weil es kurz darauf von unbekannten Personen aus den USA nach Europa gelangte, wurde gegen Zimmermann drei Jahre lang wegen Verstoßes gegen die amerikanischen Exportkontrollgesetze ermittelt, denn PGP wurde wegen seiner starken Verschlüsselung in den USA als "Waffe" eingestuft, die nicht in andere Staaten exportiert werden durfte.

Laut der Pressemitteilung vom 13.12.1999 besitzt Network Associates ab sofort eine Exportlizenz der amerikanischen Regierung zur weltweiten Ausfuhr von PGP.

Dazu Noah Salzmann von NAI am 14.12.1999:

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Hello,

I suppose I shouldn't get into a "ADK features are not Key Escrow" debate
as I am sure everyone's opinions are already set in stone...

However, I would like to address the perception (below) that the ADK
feature has been "locked-open" as this is indeed not the case.

PGP business products, as sold in the US or anywhere else, do not have
the ADK feature turned on by default. The purchasing company must turn
this on themselves.

PGP products sold in the retail channel, like Personal Privacy, _never_
force the user to respect an ADK. The retail (and freeware) users can
always encrypt to the primary key while avoiding the ADK that is
associated with that primary key. If a company wants to prevent
retail/freeware users from sending their employees ADK-less mail then
they must put up scanning tools on their mail server to enforce their
policy.

NAI/PGPinc has never entered into an agreement with the U.S. Government
in which we have traded features in PGP software for an export license,
nor would we ever do so.

We have never built a weakened version of PGP. In fact, we don't even
include 56-bit DES in our IPsec software, even though must of our
competitors do. :-P

Noah Salzman
noah@nai.com
408.346.5186

-----BEGIN PGP SIGNATURE-----
Version: PGP 6.5.2

iQA/AwUBOFayHSQ6gdj6pyJyEQK2MACgiACc/uhd6Qfjq3aNaIeIikdA8UgAn1CQ
xxMT8fRvpUoQu4YQXVOqWFME
=m+WS
-----END PGP SIGNATURE-----

Zur angesprochenen ADK Funktion siehe GAK, CMR, ARR, MRK, ADK und PGP 5/6.

1996 gründete Zimmermann seine Firma PGP Included, nachdem die Ermittlungen gegen ihn eingestellt wurden, 1997 wurde PGP Inc. von der Firma Network Associates gekauft, die noch heute mit Zimmermann zusammen PGP weiterentwickelt und vertreibt.
PGP ist ein Programm, mit dem man lokal auf der eigenen Festplatte alle Dateiarten verschlüsseln kann.
Besondere Bedeutung hat PGP aber dadurch erlangt, dass eine Vielzahl von Internetanwendern weltweit PGP zur Verschlüsselung von E-Mails und anderer Daten, die über Netzwerke versendet werden, benutzen. Darüberhinaus kann man mit PGP Texte in Klarform (Faxe, E-Mails) mit einer digitalen Signatur versehen, um auch im elektronischen Bereich, in dem eine handschriftliche Unterschrift nicht möglich ist, eine Überprüfung der Authentizität elektronisch vorliegender Texte und Daten zu ermöglichen.
Sowohl zur Verschlüsselung als auch zur Signierung setzt PGP dabei mathematische Verschlüsselungsmethoden, sogenannte kryptografische Algorithmen, wie IDEA, RSA, Diffie-Hellmann, MD5 und SHA-1, ein, die in der Welt der Kryptografie als anerkannt sicher vor Entschlüsselung, bzw. Errechnung der originalen Daten (z. B. des Klartextes einer E-Mail) aus der verschlüsselten Form durch nicht authorisierte dritte Parteien eingestuft werden.

Ein kurzer Blick auf die Struktur und Funktionsweise des Internets reicht aus, um sich die Notwendigkeit der Verschlüsselung und Signierung vor Augen zu führen. Wenn eine E-Mail versendet wird, werden die Datenpakete der E-Mail zum Mailserver des Providers übertragen, von dort versendet der Mailserver die Mail an den Ziel-Mailserver des Empfängers. Dabei wird die E-Mail meistens mehrere Rechner im Internet passieren, bis sie am Zielserver ankommt. Der Mailserver des Empfängers überträgt schliesslich die E-Mail auf den Rechner des Empfängers. Während des ganzen Transportweges werden die Datenpakete stets in lesbarem Klartext übertragen. D. h. an verschiedenen Stationen des Weges kann die E-Mail

abgefangen und auch verändert werden: Auf dem Weg vom eigenen Rechner zum Mailserver, zwischen den einzelnen Rechnern während des Transportes und vom Ziel-Mailserver zum Empfänger. Verschafft sich eine Person einen illegalen Zugang zu einem der beteiligten Rechner, kann auch dort direkt die E-Mail abgefangen werden. Zu diesem Zweck gibt es spezielle Programme wie die Packet Sniffer, mit denen Datenpakete abgefangen werden können. Die abgefangenen Pakete können auch in ihrem Inhalt verändert und wieder in den Datenstrom eingespeist werden.

Zur Verschlüsselung und Signierung bedient sich PGP des asymmetrischen Verschlüsselungsverfahrens und der Idee des "Web of Trust".

Asymmetrisches Schlüsselverfahren

Bei PGP 6 wird, wie schon zuvor mit der Version 2.6.3 ia oder 2.6.3 in, das asymmetrische Schlüsselverfahren eingesetzt. Das heisst, jeder PGP-Benutzer hat zwei zusammengehörende Schlüssel (ein Schlüsselpaar), bestehend aus:

Private Key ("privater Schlüssel")

der Aufbewahrungsort für den eigenen (wenn man nur einen PGP-Key benutzt) oder die eigenen (wenn man mehrere PGP-Keys verwendet) privaten Schlüssel ist der private Schlüsselbund (die Datei "secring.skr").
PGP bildet einen Hashwert aus der gewählten Passphrase, den "Secret Key". Mit dem Secret Key wird der Private Key von PGP verschlüsselt und dann im Secring gespeichert. Somit wird der Private Key direkt durch den Secret Key und indirekt durch die dem Secret Key zugrunde liegende Passphrase geschützt.
Detaillierte Informationen zur Passphrase, auch "Mantra" genannt, finden sich in der Mantra FAQ

Mit dem Private Key werden verschlüsselte Nachrichten entschlüsselt und die digitalen Unterschriften ("Signaturen") erzeugt.
Deshalb wird PGP jedesmal, wenn man entschlüsselt oder signiert, die Passphrase abfragen, da ja in beiden Fällen der Private Key, bzw. Secret Key zum Einsatz kommt.

Public Key ("öffentlichen Schlüssel")

der Aufbewahrungsort für alle öffentlichen Schlüssel, also auch dem eigenen Public Key ist der öffentliche Schlüsselbund (die Datei "pubring.pkr").
Mit dem Public Key wird eine Nachricht verschlüsselt oder die Signatur eines Absenders überprüft. Verschlüsselung und Signierung können dabei miteinander kombiniert werden. Dabei kann man aus dem Public Key den Private Key nicht berechnen und ohne den Private Key zu besitzen, kann man keine Nachricht entschlüsseln, die mit dem dazugehörigen Public Key verschlüsselt wurde. Auch wenn mit dem Private Key eine Signatur erstellt wurde, kann man aus dieser Signatur nicht den Private Key errechnen.
Darüberhinaus kann eine Nachricht nicht entschlüsselt, bzw. Signaturen erstellt werden, wenn der Private Key entwendet wurde, wenn der Angreifer die Passphrase und damit den Secret Key nicht kennt, bzw. über Wörterbuchangriffe (Dictionary Attack) errechnen kann.

Daraus folgt:

1. **dass der Public Key ohne Bedenken weitergeben werden kann, der Private Key niemals**
2. **dass eine starke Passphrase gewählt wird**
3. **das die Passphrase niemals weitergegeben und/oder für andere Personen zugänglich aufbewahrt wird**
4. **dass die Passphrase nie in Vergessenheit gerät**

Mehr noch:

Der Private Key sollte möglichst sicher abgelegt sein. Zum Beispiel kann man den Secring auf einer Diskette abspeichern und diese nur einlegen, wenn man PGP benutzen will. Dazu muss in PGP noch der Pfadhinweis zum Secring abgeändert werden. Fallweise ist eine zusätzliche Sicherung des Secrings durch Verschlüsselung der Datei Secring.skr oder der gesamten Diskette durch ein anderes Verschlüsselungstool überlegenswert.
Geeignete und sichere Verschlüsselungstools auf DOS und Windowsebene finden sich auf der No Big Brother Page - Kryptografische Programme.
Denn wenn jemand den Secret Key und die ihn schützende Passphrase des Keybesitzers kennen würde, könnte er alle an den Keybesitzer verschlüsselte E-Mails entschlüsseln ! Doch dazu später.

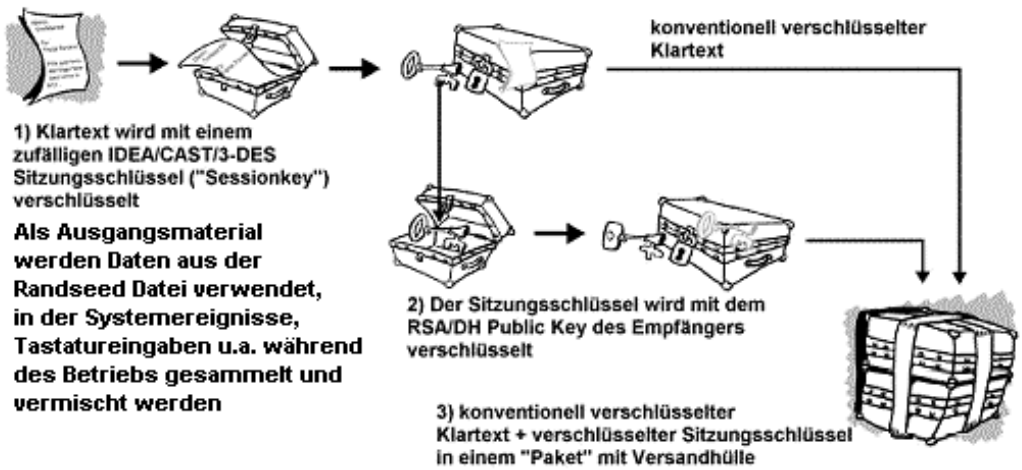
PGP und sicheres Umfeld

Es ist sinnvoll, sich draüber hinaus weitergehende Überlegungen zur Sicherheit des Systems, auf dem PGP arbeitet, zu machen, d. h. für eine möglichst sichere Umgebung zu sorgen. Das trifft gerade für Windowssysteme zu. Denn es hilft nichts, PGP zu verwenden, aber z. B. die Möglichkeit zuzulassen, dass ein Hacker über das Internet in den eigenen PC einbricht, ein Remote Control Sever über ein "Trojanisches Pferd" Virus installiert oder die Passphraseeingabe durch anwesende Personen in der Umgebung vom Monitor abgelesen werden kann.
Neben der Verwendung eines starken Keys mit einem sicheren Algorithmus und einer starken Passphrase sollte man deshalb für einen starken Virenschutz sorgen, also z. B. alle Programme vor Installation mit einem Virenscanner überprüfen, den eigenen Rechner gegenüber dem Internet mit einem Firewall- und/oder Intrusion Detection Programm absichern und sensible Daten wie den Private Key und die Passphrase sicher "verwahren". Sieht man einmal von einem TEMPEST-Angriff ab, gegen den ein normaler Anwender eh keine Chance hat, kann man so wenigstens die wichtigsten Angriffspunkte, die ein durchschnittlicher Angreifer wählen würde, neutralisieren.

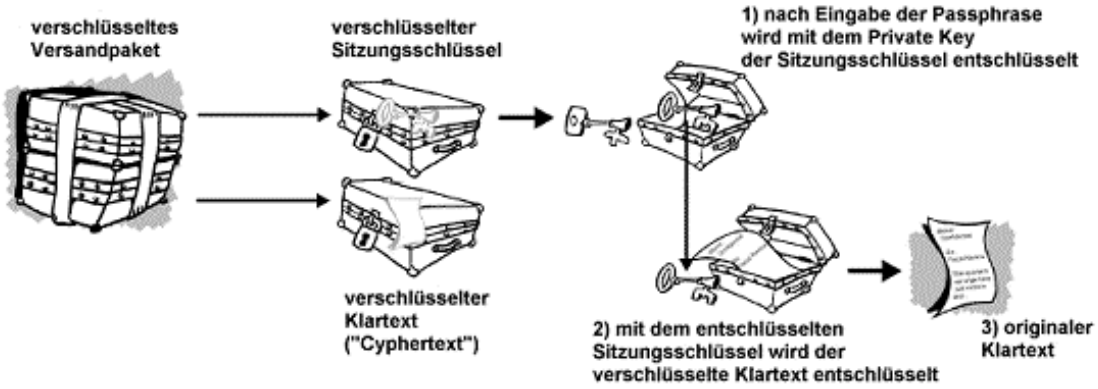
Für Windows Systeme siehe Win-Security Page und No Big Brother Page No. 2.

Grafische Darstellung des Ver- und Entschlüsselungsvorgangs beim asymmetrischen Verfahren mit PGP:

Verschlüsselung durch den Absender:



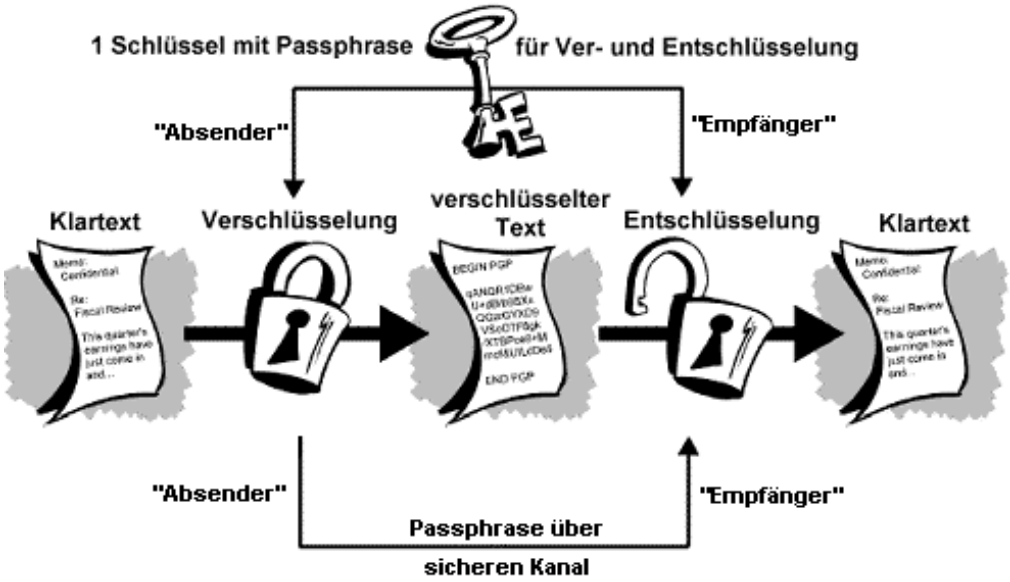
Entschlüsselung durch den Empfänger:



Wie aus den Abbildungen hervorgeht, kommt beim asymmetrischen Verschlüsselungsverfahren auf Seiten des Absenders der öffentliche RSA oder DH Schlüssel (Public Key) des Empfängers, der weltweit jeder Person, die PGP einsetzt, zur Verfügung steht und ein konevntioneller Verschlüsselungsalgorithmus wie IDEA, CAST oder Triple-DES zum Einsatz, aber kein Passwort oder eine Passphrase. Auf Seiten des Empfängers kommt der private Schlüssel (Private Key) des Empfängers, der nur ihm vorliegt und sonst keiner anderen Person, seine Passphrase und einer der konventionellen Verschlüsselungsalgorithmen zum Einsatz.

Symmetrisches Schlüsselverfahren

Grafische Darstellung des Ver- und Entschlüsselungsvorgangs beim symmetrischen Verfahren



Wie aus der Abbildung hervorgeht, wird beim symmetrischen Verschlüsselungsverfahren der Absender den Klartext, z. B. eine E-Mail, mit einem IDEA, CAST oder Triple-DES Schlüssel verschlüsseln, wobei der Schlüssel durch eine Passphrase, die sich der Absender ausdenkt, verschlüsselt, bzw. geschützt wird. Der Empfänger wiederum muss die gleiche Passphrase

angeben, damit der gleiche Schlüssel, der auf Seiten des Absenders noch zur Verschlüsselung eingesetzt wurde, als Entschlüsselungsschlüssel den verschlüsselten Text in Klartext entschlüsselt.

Damit der Empfänger aber überhaupt den Entschlüsselungsvorgang starten kann, muss ihm zuvor durch den Absender die Passphrase über einen abhör- und abfangsicheren Kanal (z. B. persönliches Treffen, abhörsichere Telefonleitung) zugegangen sein. Wenn aber ein sicherer Kanal die Vorbedingung ist, bzw. existiert, kann der Absender auch gleich den Klartext über diesen sicheren Kanal an den Empfänger weitergeben, ohne eine Verschlüsselung einsetzen zu müssen. Der Vorteil beim asymmetrischen Verfahren besteht also gerade darin, dass dieser sichere Kanal nicht vorliegen muss, um Texte abhör- und abfangsicher über grosse Entfernungen austauschen zu können, da auch der Austausch einer Passphrase keine Vorbedingung darstellt.

Aus diesem Grund ist das symmetrische Verfahren nur sinnvoll, wenn es entweder von ein und derselben Person verwendet wird, z. B. vom Anwender, der seine Dateien auf der eigenen Festplatte verschlüsseln möchte (so wird das symmetrische Verfahren von PGP und PGPdisk zur Verschlüsselung lokaler Daten angewendet) oder in einem Umfeld, wo Daten für zwei Personen verfügbar vorliegen sollen, aber keiner dritten Person und diese zwei Personen die Möglichkeit haben, die gemeinsame Passphrase persönlich auszutauschen, z. B. zwei Kollegen, die ein gemeinsames Projekt in einer Firma bearbeiten.

Wer sich einen näheren Überblick zu kryptografischen Verfahren und Begriffen verschaffen möchte, sollte sich die (englischsprachige) FAQs About Today's Cryptography? der RSA Labors besorgen.
Zu beachten ist auch, dass sich das Dateiformat des Pub- und Secrings von PGP 6 von dem Dateiformat von PGP 2.6.3 unterscheidet. Deshalb sollte man darauf achten, dass ein Zugriff einer PGP Version auf die falsche Datei ausgeschlossen ist und es nicht zu einer versehentlichen Umbenennung oder Löschung einer Pub- oder Secring Datei kommt, wenn sich die Dateien beider Versionen im gleichen Verzeichnis befinden oder bei beiden Versionen der gleiche Zugriffspfad angegeben ist. Veränderungen an den Public- und Private Keys, die in einer PGP-Version durchgeführt werden, sollten analog in der anderen PGP-Version wiederholt werden. Auf diesem Wege können Fehlermeldungen, die beim Ent- und Verschlüsseln aufreten und fälschlicherweise einer Inkompatibilität oder Keyfälschung zugeschrieben werden, im Vorfeld verhindert werden.

Verschlüsselungs- und Signieralgorithmen oder
"RSA und Diffie-Hellmann PGP Keys, MD5, DSS und SHA-1"

Weiter oben wurde ja bereits erläutert, was es mit "Secret" und "Public" Key auf sich hat. Es gibt zwei Schlüsseltypen bei PGP 5/6, die nach verschiedenen Verschlüsselungsalgorithmen, also grob gesagt mit verschiedenen Berechnungsschemata, arbeiten.
Die RSA - und die Diffie-Hellmann/DSS - Public Keys

RSA-Keys mit IDEA und MD5

Die Bezeichnung des Algorithmus "RSA" setzt sich aus den Anfangsbuchstaben der Namen seiner Erfinder Rivest, Shamir und Adleman zusammen.
RSA wird seit langer Zeit zur Verschlüsselung eingesetzt, so auch in den Vorgängerversionen von PGP und gilt als gut untersucht und geprüft.
IDEA ist die Abkürzung für "International Data Encryption Algorithm", einem symmetrischen/Single-Key (konventionellen) Verschlüsselungsalgorithmus.
Beide Algorithmen werden bei PGP zur Verschlüsselung kombiniert eingesetzt.
MD5 steht für "Message Digest 5" einem weiteren Algorithmus, mit dem eine 128-bit lange Textprüfsumme eines Textes erstellt wird. Wenn diese Textprüfsumme mit dem Secret Key codiert wird, entsteht die digitale Signatur von PGP.
RSA Keys haben eine Schlüssellänge von 516 bis 16384 bits, die eingesetzte Standardlänge liegt bei 2048-bit.

Diffie-Hellmann/DSS-Keys mit IDEA, CAST, Triple-DES und SHA-1

Auch hier wurden die Namen der Erfinder Diffie und Hellmann zur Bezeichnung herangezogen.
DSS ist die Abkürzung vom "Digital Signature Standard".
Dieser Standard wurde als Teil des "Capstone"-Programmes (ein weiterer bekannter Bestandteil dieses Programmes war der Clipper-Chip) der US-Regierung vom NIST ("National Institute of Standards and Technology", eine Abteilung des amerikanischen Handelsministeriums) in Zusammenarbeit mit der NSA (die "ofiziell" eine "beratende" Funktion beim NIST ausübt) zum digitalen Beglaubigungsstandard der US-Regierung. Mit dem DSS Standard wurde der "Digital Signature Algorithm" (DSA) als offizieller Signieralgorithmus eingeführt.
DSS/DH-Keys werden seit der PGP-Version 5.0 eingesetzt. Sie haben eine Gesamtschlüssellänge von 512 bis 8192 Bits, die eingesetzte Standardlänge liegt bei 4096-bit, wobei der DSS-Anteil, wie in den DSA Spezifikationen festgelegt, immer 1024 Bits beträgt. DSS ist der Public Key-Anteil, der für die Signierung zuständig ist. Seit PGP 6 können ein separater DSS Master Signing Key und mehrere DH Encryption Keys (die sog. Subkeys) erstellt werden.
SHA-1 steht für "Secure Hash Algorithm 1", der bei PGP 5/6 die 160-bit lange Textprüfsumme erstellt, die nach Verschlüsselung mit dem Secret Key die digitale Signatur erzeugt.
Neben IDEA kann man mit PGP 5/6 auch zwei weitere, symmetrische Algorithmen einsetzen: CAST steht für die beiden Erfinder Carlisle Adams and Stafford Tavares, die den Algorithmus für die Northern Telecom (Nortel) entwickelten und Triple-DES ist eine Abwandlung des *Data Encryption Algorithm (DSA)* des noch gültigen *Data Encryption Standard (DES)*, der 2001 durch den Advanced Encryption Standard (AES) abgelöst wird.
Zu Diffie-Hellmann ist die Forschung nicht so umfangreich wie zu RSA, der SHA-1 Algorithmus gilt aber als sicherer als MD5, der einige Schwächen aufgezeigt hat.

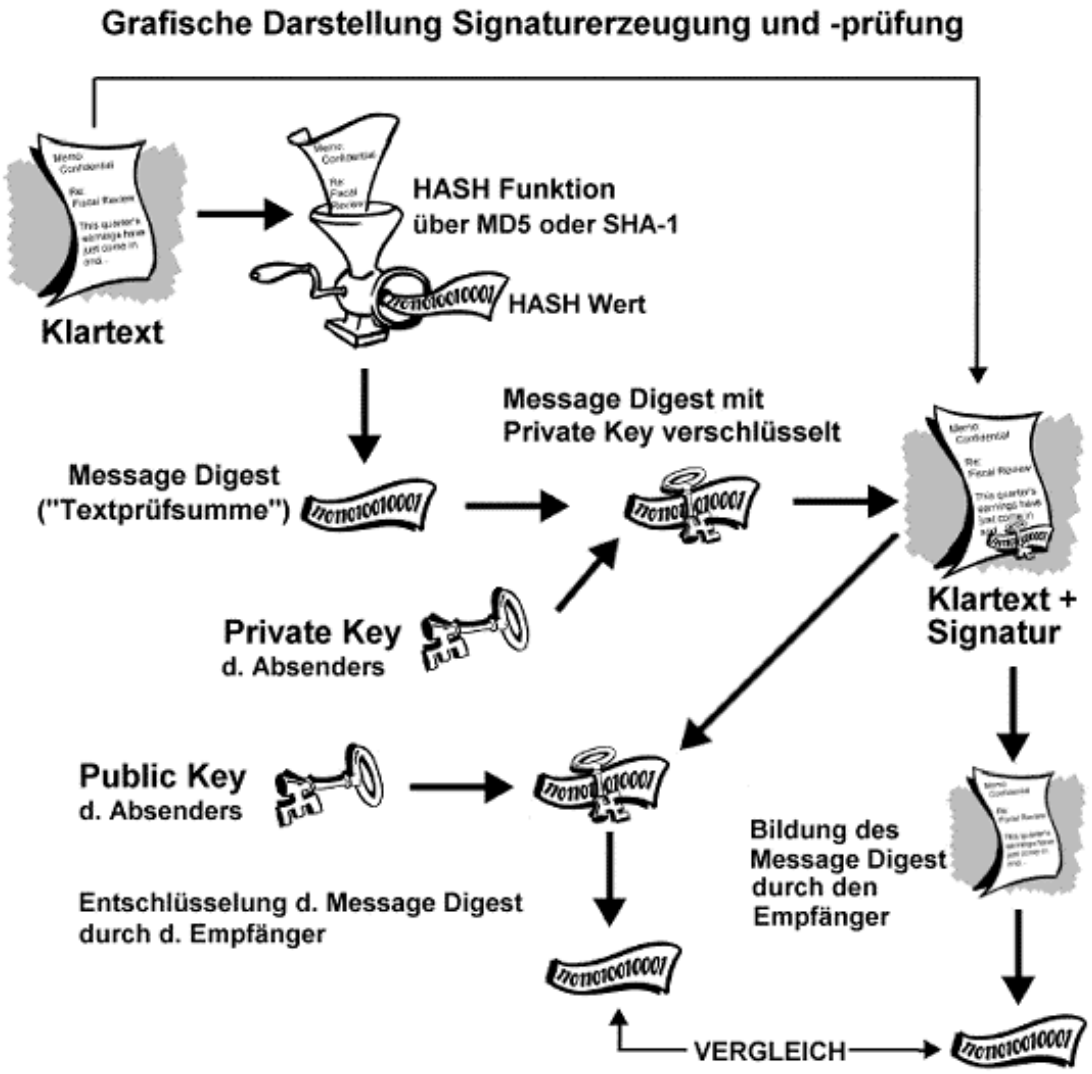
Wer mehr zu den Hintergründen der beiden Schlüsseltypen erfahren möchte, dem sei die PGP DH vs. RSA FAQ von Sam Simpson empfohlen.

Stichworte Digitale Signatur, Hash, Message Digest

Zur Erzeugung einer *digitalen Signatur* wird zu einem Ursprungstext von beliebiger Länge über eine mathematische Transformation, der *Hash Funktion*, eine Zeichenkette mit feststehender Länge berechnet und komprimiert, der so erzeugte *Hashwert* wird *Message Digest* oder auch *Textprüfsumme* genannt.
Zur Erzeugung des Message Digests werden bei PGP die Message Digest Algorithmen MD5 und SHA-1 eingesetzt (ein weiterer A. ist RIPE-MD 160).
Da der Message Digest stellvertretend für den Ursprungstext steht, spricht man auch vom *digitalen Fingerabdruck* eines Dokuments. Dabei kann aus dem Message Digest der Ursprungstext nicht wieder zurückberechnet werden und bei starken und sicheren Hashfunktionen muss der Hashwert pro Dokument einzigartig sein, d. h. eine Hashfunktion darf nicht für zwei unterschiedliche Texte ein und denselben Hashwert erzeugen (können). Anschliessend wird der Message Digest mit dem RSA oder DSS/DSA Private Key des Absenders verschlüsselt, das Resultat nennt man die *digitale Signatur* eines Dokuments.

Auf der Empfängerseite bildet der Empfänger mit der gleichen Hash Funktion ebenfalls den Message Digest zum Text. Anschliessend entschlüsselt er mit dem Public Key des Absenders den vom Absender gebildeten Message Digest aus der digitalen Signatur und vergleicht beide Message Digests miteinander. Sind beide Hashwerte gleich, ist das die Bestätigung, dass die digitale Signatur dem Text entspricht, also echt ist und die Signatur mit dem Secret Key des Absenders erstellt wurde, also der Text von Absender XY stammt, wenn man voraussetzt, dass der Key wirklich zum Absender XY gehört.

Erzeugung und Überprüfung digitaler Signaturen



Bemerkung:
Besonders interessant bei der ganzen Geschichte um Kryptographie, Verschlüsselungsverfahren und Schlüsseltypen ist die Rolle der "National Security Agency" (NSA), einem streng geheimen Geheimdienst der USA, der über ein Höchstmass an technischem Equipment und Experten Know-How verfügt und von Beginn an in die Entwicklung, das Testen und Brechen von Verschlüsselungsverfahren involviert war und ist. Er stellt die mehr oder weniger grosse Unbekannte bei der Verwendung von kryptographischen Programmen dar. Nähere Infos zur NSA über EPIC oder direkt bei der NSA.

Nun gut, wer sich mehr für die mathematischen Prozeduren und genauen Verschlüsselungsabläufe interessiert sei auf andere Quellen verwiesen.

Kompatibilitätsvergleich zwischen PGP 5/6 und PGP 2.6.3

Von PGP 2.6.3 zu PGP 5/6

- Da PGP 5/6 abwärtskompatibel ist, kann es RSA-Keys, die mit PGP 2.6.3 erstellt wurden in seine Keyrings aufnehmen

Je nach verwendeter PGP 2.6.3 Version können das 2048, 4096, 8192-bit grosse Keys sein
- Ebenso können mit PGP 2.6.3 verschlüsselte oder/und signierte Daten von PGP 5/6 entschlüsselt und überprüft werden
- Wenn man zu PGP 5/6 wechseln will, kann man den Public und Secret Key aus den Pub- und Secring von PGP 2.6.3 mit dem Befehl

```
PGP 2.6.3> pgp -kxa Benutzer-ID Keydateiname.asc pubring\secring.pgp
```

extrahieren und dann in den Pub- und Secring von PGP 5/6 übernehmen

Von PGP 5/6 zu PGP 2.6.3

- Ein RSA-Public Key, der mit PGP 5/6 erstellt wurde, kann in den Pubring von PGP 2.6.3 aufgenommen werden
- Daten, die mit einem RSA-Key mittels PGP 5/6 verschlüsselt und/oder signiert wurden, können mit PGP 2.6.3 entschlüsselt und überprüft werden
- Ein DSS/DH-Key kann nicht in den Pubring von PGP 2.6.3 aufgenommen werden
- Daten, die mit einem DSS/DH-Key verschlüsselt und/oder signiert wurden, können nicht mit PGP 2.6.3 entschlüsselt und überprüft werden
- Wenn man von PGP 5/6 zu PGP 2.6.3 wechseln will, kann man den Secret Key aus PGP 5/6 auf folgende Weise in den Secring von PGP 2.6.3 übernehmen:
    1. das mit PGP 5/6 erzeugte Schlüsselpaar markieren
    2. Menü "Keys", Funktion "Export"
    3. im Exportwindow, die Option "Export private Key(s)" aktivieren und die Keys als Datei speichern
    4. im Exportwindow, die Option "Include 6.0 Extensions" **nicht** aktivieren
    5. die erzeugte Keydatei mit einem Editor (z. B. Notepad) öffnen
    6. den Eintrag für den Public Key (Begin Public Key Block) löschen
    7. den Secret Key mit dem Befehl

```
PGP 2.6.3> pgp -ka Secret Key Dateinamen Pfad:\secring.pgp
```

in den Secring von PGP 2.6.3 aufnehmen

Beispiele

(bei denen der Empfänger PGP 2.6.X) benutzt:

- E-Mail wird mit dem eigenen DSS/DH-Secret Key signiert
  = Empfänger kann Signatur nicht prüfen
- E-Mail wird mit RSA-Key des Empfängers verschlüsselt und mit DSS/DH-Secret Key signiert
  = Empfänger kann E-Mail entschlüsseln, aber die Signatur nicht prüfen
- E-Mail wird mit RSA-Key des Empfängers verschlüsselt und gleichzeitig mit dem DSS/DH-Key des Absenders
  = Empfänger kann weder die E-Mail entschlüsseln, noch die Signatur prüfen

GnuPG und PGP 6.X

DH/DSS Keys, die mit PGP 6.X erzeugt wurden, können in die Keyringe von GnuPG importiert werden. Dazu muss das Keypaar zuerst über PGPkeys exportiert werden. Anschliessend importiert man beide Keys über die Befehle

```
gpg> gpg  --import  Keydateiname
```

```
gpg> gpg  --import  --secret-key Keydateiname
```

Nachteile und Sicherheitsrisiken von PGP 5/6 gegenüber PGP 2.6.3

Ein gravierender Nachteil von PGP 5/6 ist der Wegfall einiger Funktionen, die mit PGP 2.6.3 noch möglich sind und waren:

- Das Verhindern der Abspeicherung der entschlüsselten Daten mittels (womit nur die Anzeige der entschlüsselten Daten auf dem Monitor erreicht wird, um dem Enpfänger den Hinweis zu geben, die entschlüsselten Daten nicht abzuspeichern) gibt es bis zur Version PGP 6.0 nicht, ab Version 6.0.2 ist diese Funktion wieder verfügbar:

  ```
  PGP 2.6.3> pgp -sem Datei
  ```

- Darüberhinaus ist eine Veränderung des Sourcecodes in Form von Bugfixes oder Patches zum Zweck der Fehlerbeseitigung oder Verbesserung von PGP 5/6 aufgrund lizenzrechtlicher Bedingungen untersagt.
  Es gibt aber trotzdem in Form der Cyber Knight Templar (CKT) verbesserte und fehlerbereinigte Versionen.
- Man kann bis Version 6.5.1 PGP nicht für automatisierte Vorgänge, wie z.B. als Bestandteil einer Batchverarbeitung einbinden. Erst ab Version 6.5.1 ist auch wieder eine Kommandozeilenversion dabei, die allerdings nur mit DH/DSA Keys arbeitet.
- Durch das "Feature" von PGP 5.X die sensible Passphrase bis zu 5 Minuten im Arbeitsspeicher aufzubewahren, kann es möglich sein, dass sie in der Auslagerungsdatei verbleibt, bzw. dort abgespeichert wird. Ab Version PGP 6.0 soll der enthaltene "Memlock"-Treiber dafür sorgen, dass keine sensiblen Daten in den Speichern verbleiben. Eine Abhilfe soll durch das Setzen der Zeitspanne auf 1 Sekunde erreicht werden.

  Eine sicherere Methode, nicht nur, um eine möglicherweise in der Swapdatei verbliebene Passphrase zu löschen, sondern auch ver- oder entschlüselte Daten auf der Festplatte sicher zu löschen, ist die Anwendung von Tools, die einfach und komfortabel Verzeichnisse oder ganze Partitionen verschlüsseln, oder in der Lage sind, einzelne Daten oder weite Bereiche auf der Platte ganz zu löschen und anschliessend mit Zufallszahlen oder -daten mehrfach zu überschreiben.
  Eine Auswahl findet sich auf der **No Big Brother Page - File Wiping Tools**

- Bei Verwendung der PGP 5 Kommandozeilenversion unter Linux werden keine Zufallszahlen mehr gebildet, wenn die Datei "randseed.bin" umbenannt und/oder abweichend vom Standardprogrammverzeichnis in ein anderes Verzeichnis verschoben und die Konfigurationsdatei dementsprechend angepasst wird.
  Als Ergebnis werden alle E-Mails mit dem gleichen Session Key verschlüsselt.
  Aus diesem Grund darf die "randseed.bin" bei der Kommandozeilenversion nicht verändert werden.
- Laut Draft der OpenPGP Gruppe werden in den PGP 5 Versionen die Zufallszahlen des Message-ID Headers, wie er bei jeder Verschlüsselung ("Encryption") erzeugt wird, aus dem Zufallszahlenpool, der auch für die Erzeugung des Session Keys benutzt wird, entnommen, wodurch ein Angriff auf den verschlüsselten Text erleichtert wird.

  Zitat:

  > "The Message ID is usually derivated from the same "randomness"
  > pool the session key comes from. So the MessageID header

> introduces an unnecessary security weekness, because the confidentiality of the message may be attacked by attacking the "randomness" pool functions. This header must not generated and should produce a warning to the recipient."

Abhilfe:
nach (Signierung und) Verschlüsselung eines Textes wird anschliessend die Message-ID entfernt, d. h. die dritte Zeile (beginnt mit "Message-ID: ..."), so dass weiterhin zwischen dem PGP-Block und der Zeile mit der Versionsangabe nur eine Leerzeile verbleibt.
Die Message-ID wird in den späteren 6er Versionen nicht mehr gebildet.

- Mit PGP 5/6 Businessversionen ist es möglich, PGP Anwender in geschlossenen Benutzerkreisen zur Verwendung von ADK (Additional Decryption Keys) Public Keys zu zwingen, d. h. Public Keys, die Funktionen enthalten, die eine Drittverschlüsselung an weitere Personen bewirken. Mehr dazu auf der nächsten Seite.

  Wer lieber mit PGP 2.6.3 arbeiten und dessen Features nutzen möchte, aber den gleichen Komfort und Funktionalität von PGP 5/6 benötigt, dem sei das Sharewareprogramm **PGPClick** von Robert E. Wilson (das für Star Trek Trading Cards oder 5 $ registriert werden kann) empfohlen.

GAK, CMR, ARR, MRK, ADK und PGP 5/6

Abkürzung	Übersetzung	Bedeutung
GAK	Government Access to Keys	staatlicher Zugriff auf Public Keys
ARR	Additional Recipient Record	ein Feld in einem Public Key, in dem vermerkt wird, an welchen Dritten zusätzlich verschlüsselt wird
CMR	Corporate/Company Message Recovery	Wiederherstellung des Klartextes einer verschlüsselten Nachricht durch Firmen und Vereinigungen
CMRK	Corporate/Company Message Recovery Key	der Drittkey
MRK	Message Recovery Key	der Drittkey
ADK	Additional Decryption Key	der Drittkey

Mit PGP 5 wurde eine neue Funktion eingeführt, die als CMR, "Corporate Message Recovery" bekannt wurde und eine breite Diskussion auslöste, die noch heute andauert.
CMR beschreibt die Möglichkeit, Dateien oder Nachrichten, die von einer Person an eine zweite verschlüsselt werden, gleichzeitig für eine dritte Person zu verschlüsseln, die sie also auch wieder entschlüsseln kann. Deshalb spricht man in diesem Zusammenhang auch von Drittkeyverschlüsselung. NAI hat CMR als Aternative zum "Key Recovery"-Konzept, d. h. der Rückgewinnung des eigenen Entschlüsselungskeys durch Dritte und dem Konzept des "Key Escrow", dem Hinterlegen des Entschlüsselungskeys bei einer zentralen Stelle, entworfen. Faktisch bleibt die Wirkung dieselbe: Eine dritte Partei kann in die Lage versetzt werden, die verschlüsselte Kommunikation zwischen "Bob" und "Alice" mitzuverfolgen oder verschlüsselte Dateien von "Bob" zu öffnen.

Die Funktionsweise

Der Administrator, von NAI bezeichnenderweise als "Chief Security Officer" (CSO) bezeichnet, erzeugt mit Hilfe des PGPadmin Tools eine Clientversion von PGP 5/6, die jeder Benutzer zu verwenden hat.
Während der Clienterzeugung kann der Administrator im Konfigurationsprozess bestimmte Präferenzen setzen:

- Festlegung eines ADK für eingehende E-Mails
- Festlegung eines ADK für ausgehende E-Mails
- Erzwingung der Benutzung eines ADK für ausgehende und/oder eingehende E-Mails
- Erzwingung von zusätzlichen ADKs (z. B. weiterer "Firmen")
- Erzwingung einer bestimmten Länge und Qualität der Passphrase
- Erzwingung einer bestimmten Keygrösse
- Zwang aller Benutzer der Clientversion, den ADK bei der Erzeugung eigener Schlüssel zu signieren
- Erzwingung eines bestimmten Kommentars im Nachrichtenheader
- Ausgabe einer Warnung, wenn mit einem Public Key verschlüsselt wird, der nicht mit dem ADK signiert wurde
- Verbot der Keyerzeugung
- Verbot der RSA-Keyerzeugung
- Verbot der konventionellen Verschlüsselung
- Voreinstellung der Keys, die im Standardkeyring enthalten sind

Erhält ein Benutzer die Clientversion und erzeugt damit seine Public Keys, wird diesem ein CAF (Corporate Access Field), bzw. ARR (Additional Recipient Record) Bereich hinzugefügt. Er enthält die Information, dass bei jeder Verschlüsselung sowohl mit dem Public Key des Empfängers, aber auch mit dem Firmen Public Key (CMRK/ADK) verschlüsselt werden "soll" oder "muss", bzw. bei jeder Verschlüsselung der Firmen Public Key (CMRK/ADK) verwendet wurde.
Der Public Key des Benutzers wird so zum Träger des CMRK/ADK der Firma: Aufgrund der Zusatzverschlüsselung ist der Firmenzugriff auf den Session Key (und damit auf den Nachrichtentext) gewährleistet.

Man unterscheidet drei Arten von ADK´s:

1. ADK´s für eingehende Dateien/Nachrichten
   wird der Public Key eines Benutzers von einer Person ausserhalb der Firma benutzt, wird nicht nur mit dessen Key verschlüsselt, sondern auch mit dem Drittkey. Diese Möglichkeit kann nur für DH/DSS Keys konfiguriert werden.

2. ADK´s für ausgehende Dateien/Nachrichten
   verschlüsselt ein Benutzer in der Firma an einen anderen Benutzer in der Firma oder an eine Person ausserhalb der Firma, wird auch mit dem Drittkey verschlüsselt. Diese Möglichkeit kann für DH/DSS und RSA Keys konfiguriert werden.

3. ADK´s dritter Firmen
   die Clienten werden so konfiguriert, dass der Benutzer mit einem Drittkey verschlüsseln muss, wenn er den Public Key mit ADK einer dritten Partei erhält und diesen zur Verschlüsselung benutzen will.

Deshalb ist es für eine Firma, die zwingende Drittkeyverschlüsselung für beide Kommunikationsrichtungen einsetzen will sinnvoll, nur DH/DSS Keys zuzulassen und die Clienten so zu konfigurieren, dass ADK-Verschlüsselung zwingend ist, die RSA-Keyerzeugung für den Benutzer nicht zu erlauben und die Möglichkeit der konventionellen Verschlüsselung für die Benutzer auszuschalten.

Um die Drittkeyverschlüsselung innerhalb eines geschlossenen Benutzerkreises wie einer Firma lückenlos zu überwachen, kommen in diesem Konstrukt noch drei weitere Komponenten hinzu:

1. PGP Certificate Keyserver
   Um ADK-Keys zu verhindern, könnte ein Benutzer versuchen, mit einer unabhängigen PGP Version einen Public Key zu erzeugen und ihn in der Firma in Umlauf zu bringen. Um dies zu verhindern, richtet der Administrator einen Corporate Signing Key (CSK) ein, mit dem alle neuen Public Keys signiert sein müssen, bevor sie vom PGP Keyserver akzeptiert werden.
   Erzeugt ein Benutzer einen unabhängigen Key, der nicht den Sicherheitsrichtlinien (Policy) entspricht (weil ihm z.B. der ADK fehlt), wird die Signatur vom Keyserver versagt, der Public Key zurückgehalten und gerät nicht in Umlauf.
   Umgekehrt signieren alle ADK-Keys der Benutzer den CSK und werden wiederum vom CSK signiert. Der PGP Client kann so konfiguriert werden, dass eine Warnung an den Benutzer erfolgt, wenn an einen Key verschlüsselt wird, der nicht vom CSK signiert wurde und der PMA (siehe unten) kann so eingerichtet werden, dass Daten, die mit einem Public Key ohne CSK Signatur verschlüsselt wurden, abgeblockt und nicht weitergeleitet werden.

2. Policy Management Agent for SMTP (PMA)
   Da mit Einsatz des PMA die Einhaltung der Sicherheitsrichtlinien (Policy) (auch der Drittverschlüsselung) durchgesetzt wird, wird der PMA in Gemeinschaft mit dem SMTP Server auch als "Policy Enforcer" bezeichnet.
   Der PMA arbeitet mit jedem SMTP Mailserver zusammen und führt folgende Kontrollfunktionen aus:

   - der ARR-Bereich wird überprüft, ob eine Drittkeyverschlüsselung vorliegt oder nicht. Fehlt die Drittkeyverschlüsselung, wird die Nachricht nicht versendet und an den Absender zurückgeschickt.
   - Nachrichten, die mit einem Public Key verschlüsselt oder mit einem Key signiert wurden, der nicht vom CSK signiert ist, werden abgeblockt.
   - Nachrichten, die konventionell verschlüsselt wurden, werden nicht versendet und an den Absender zurückgeschickt.
   - Nachrichten, die mit bestimmten Keys verschlüsselt wurden oder an bestimmte Adressen gehen sollen, die nicht zulässig sind, werden nicht versendet und an den Absender zurückgeschickt.

3. Designated Revoker
   In der PGP Version 2.6.3 und PGP bis Version 5.5 kann nur der Besitzer des Public Keys diesen auch wieder zurückziehen, seit der Version PGP 6 gibt es eine weitere Person, bzw. dessen Key, der einen anderen Public Key zurückziehen und damit ungültig machen kann.
   Ein Designated Revoker ist ein weiterer Benutzer, der dazu ermächtigt ist, den eigenen Public Key zu widerrufen, bzw. auf dem Keyserver als zurückgezogen zu kennzeichnen.
   Der PGP Client kann vom Administartor so konfiguriert werden, dass für jeden Public Key, der mit der Clientversion erzeugt wurde, ein Revoker Key (z.B. der Key des CSO) vorhanden ist. Der Designated Revoker Key muss ein DH/DSS Key sein. Verstösst ein Benutzer gegen Policies oder verliert das Vertrauen der Firma, kann also eine dritte Person den Public Key des Benutzers widerrufen, so dass der Benutzer nicht mehr in der Lage ist, mit diesem Key verschlüsselte Nachrichten zu lesen.

Zur Verdeutlichung kann man sich diese CMRK-Keys herunterladen und in den Pubring aufnehmen.
Der eine DH-Key von "CMR-User" stellt den User dar, der einen Public Key mit Drittverschlüsselung besitzt, der DH-Key von User "Little Brother" ist der Drittkey an den mitverschlüsselt wird.

Die Beschreibung der Drittkeyverschlüsselungs- und Kontrollfunktionen in PGP 5/6 legen die Vermutung nahe, dass die neuen DH/DSS-Keys nicht bloss deshalb eingeführt wurden, weil es sich um einen neuen Algorithmus handelt oder aufgrund lizenzrechtlicher Überlegungen (denn PGP 6.0 for Business Security unterstützt sehr wohl RSA), sondern auch, weil die DH/DSS Keyinfrastruktur innerhalb geschlossener Benutzergruppen vielfältigere Kontroll- und Zugriffsmöglichkeiten bietet.

Schutzmöglichkeiten bei PGP 5/6

Generell sollte nur eine PGP Version eingesetzt werden, die ADK/CMRK Keys im PGPkeys Fenster anzeigen kann. Dazu ruft man PGPkeys auf und aktiviert im Menü **View** den Menüpunkt **ADK** (PGP 6.0.2).
Nicht-CMRK-Keys werden in der ADK Spalte mit grauen Buttons, CMRK-Keys mit einem roten Button markiert, so dass eine leichte Identifizierung von CMRK-Keys möglich ist. Leider

wird der verantwortliche Drittkey nicht in gleicher Weise gekennzeichnet. Um die ADK Anzeige zu optimieren, sollte man zuerst alle Anzeigeoptionen deaktivieren und dann reaktivieren, wobei die ADK Anzeige an erster Stelle aktiviert wird, so dass die ADK Anzeigespalte sofort hinter der Keys Spalte angefügt wird.
An einer weiteren Stelle fallen CMRK-Keys auch sofort auf: Wird an einen CMRK-Key verschlüsselt, in dem der PGP Key des Empfängers in das Recipients Fenster gezogen wird, erscheint automatisch auch der Drittkey, an den mitverschlüsselt wird. Ist der Drittkey im Pubring nicht vorhanden oder deaktiviert (disabled) worden, erscheint bei dem Versuch, an den CMRK-Key zu verschlüsseln die Warnmeldung:

The user XYZ has a missing Additional Decryption Key. Contact the owner of the key to obtain the Additional Decryption Key.

Ein weiteres, eindeutiges Indiz, dass ein CMRK-Key vorliegt. In diesem Fall sollte man Kontakt mit dem E-Mailpartner aufnehmen, einen anderen E-Mailweg und einen anderen PGP-Key vereinbaren.

Es soll trotzdem an dieser Stelle deutlich darauf hingewiesen werden, dass ein zu Hause vom Privatuser mittels einer PGP 5/6 Version (die aus sicherer Quelle stammt) erzeugter Key genauso sicher ist wie ein Key, der mit PGP 2.6.3 erzeugt wurde. Die ungewollte Drittkeyverschlüsselung über den eigenen Key ist ausgeschlossen. Bekommt man aber von einem Mailpartner einen DH-Key, der mit zwingender Drittkeyverschlüsselung ausgestattet ist, wird der Mailpartner die Nachricht nur erhalten können, wenn auch an den Drittkey verschlüsselt wurde.

Die gefährlichen Möglichkeiten

"Privacy" ist in einem geschlossenen Benutzerkreis, in dem PGP 5/6 eingesetzt wird, nicht existent.

Im Rahmen einer gesetzlichen Regelung zur Anwendung kryptografischer Verfahren (z. B. als Ergänzung zum TKG), wären Vorschriften denkbar, die Provider und Onlinedienste dazu verpflichten, den PGP SMTP Server (oder einen Server mit vergleichbarer Funktionalität) einzusetzen und von den Benutzern fordert, nur Public Keys mit CAF/ARR und staatlich/behördlicher CMR-Funktion einzusetzen.
Aus dem "Firmen"-Master-Key würde der Behörden-Master-Key, aus Corporate Message Recovery würde "Government Message Recovery", und aus dem Corporate Access Field würde ein "Law Enforcement Access Field" werden. Wenn so der geschlossene Benutzerkreis einer Firma durch die Bürger eines Staates ersetzt würden, wäre der Begriff "Privacy" im Internet nicht mehr anwendbar, da der Staat zwar kein GAK ("Govermental Access to Keys") hätte, dafür aber ein GAM ("Governmental Access to Messages").
Die Firma PGP.INC (jetzt Network Associates NAI, alias McAfee) hat Firmen, Ministerien, staatlichen Behörden und Diensten mit ihrem Produkt PGP 5/6 die technischen Möglichkeiten aufgezeigt und technischen Mittel in die Hand gelegt, um eine effektive und umfassende Überwachung des E-Mailverkehrs innerhalb eines begrenzten Raumes wie auch auf nationaler Ebene Wirklichkeit werden zu lassen und damit der ganzen Bewegung für sichere E-Mailverschlüsselung und das Recht auf Privacy schweren Schaden zugefügt. Das hat mit den ursprünglichen Ideen und Idealen, die sich mit Pretty Good Privacy verbinden, nichts mehr zu tun.

NAI, TIS, PGP, KRA und Key Recovery

Anfang Dezember 1997 wurde PGP für 35 Millionen US Dollar an Network Associates (NAI) verkauft. Zu diesem Zeitpunkt war NAI schon Mitglied in der Key Recovery Alliance.

Die Key Recovery Alliance K R A

der im Oktober 1996 gegründete Unternehmensverband (zu dem z. B. AOL, Compaq, Apple, DSI, Entrust, Fujitsu, Hewlett-Packard, IBM, Mitsubishi, Motorola, NEC, Novell und VeriSign gehören) hat sich zum Ziel gesetzt, die Entwicklung, Implementation und den Aufbau einer globalen Infrastruktur von Technologien zu fördern, die es dritten Parteien ermöglicht, über Key Recovery (Möglichkeit einer dritten Partei den Secret Key wiederherzustellen) und Key Escrow (Hinterlegung des Secret Keys an zentraler Stelle) verschlüsselte Daten zu rekonstruieren. Die KRA stimmt damit mit der Politik der US Regierung überein, den Export starker Kryptografie zu verbieten oder zu verhindern, es sei denn, sie beinhaltet Vorkehrungen (Backdoors), die es amerikanischen Geheimdiensten ermöglicht, die verschlüsselten Daten wiederherzustellen. Gleichzeitig versucht sie, einem staatlich vorgeschriebenen, zwingendem Key Recovery (mandatory key recovery) durch ein marktwirtschaftlich begründetes Key Recovery zuvorzukommen.

Neben den technischen Vorstellungen stellt die KRA auch politische Überlegungen zu einem staatlichen Key Recovery Szenario (GAK) an:

die KRA meint, dass die Politik Key Recovery Gesetze verfassen muss, in denen festgelegt wird:

- unter welchen Umständen staatliche Behörden gesetzmässig auf die Rekonstruktion verschlüsselter Daten zugreifen dürfen
- dass ein Key Recoveryzugriff nur unter gerichtlicher Aufsicht und nach gerichtlicher Anweisung erfolgen darf und der Zugriff streng an die gerichtlichen Vorgaben gebunden ist
- wie die Kontrolle und Dokumentation über Aufbewahrung und/oder Vernichtung rekonstruierter Daten geregelt ist
- dass der Dateninhalt nach seiner Rekonstruktion nicht durch staatliche Einwirkung verändert wird

Nach dem Verkauf von PGP.INC an NAI nimmt Phil Zimmermann bei NAI die Funktion eines Beraters/Mitarbeiters ein, während Phil Dunkelberger, ehemaliger Präsident/CEO von PGP.INC, bei NAI General Manager der Total Network Security Abteilung (zu der PGP gehört) wird.
Während Phil Zimmermann zu Key Recovery einmal gesagt hat, dass Key Recovery Funktionen die Hand eines Polizeistaates stärken würde und eine Invasion der Privatsphäre darstelle, erwähnte er später zur CMRK Funktion in PGP 5/6, dass, solange er etwas dazu zu sagen habe, diese Funktion für die Benutzer optional sei und der Absatzmarkt für PGP solche Funktionen verlange.
Am 6. Dezember 1997, ca. eine Woche nach dem Verkauf, verkündet NAI den Ausstieg aus der KRA mit der Begründung, die Position von NAI und PGP sei es, Regierung und Industrie dazu zu ermutigen, zu einer Politik überzugehen, die den Export starker Kryptografie ohne erzwungene Key Recovery erlaube.
Zur KRA lässt der Direktor des NAI Produktmanagements, Gene Hodges, verlauten, dass das technische Interesse an Key Recovery verständlich, bei NAI aber die Hoffnung bestünde, dass zwingende Key Recovery nicht das Ziel der Regierungspolitik sei. Es ist anzunehmen, dass dieser Schritt von NAI auf Druck von Phil Zimmermann und vieler Stimmen aus der Kryptoszene unternommen wurde um die Reputation von PGP zu schützen, zumal Zimmermann und die ehemalige PGP.INC Crew von der Mitgliedschaft NAI´s in der KRA wohl erst aus einer Meldung der WIRED NEWS erfahren hatte.
Am 25. Februar 1998 kauft NAI für 300 Millionen US-Dollar die Firma Trusted Information Systems (TIS), die enge Verbindungen zur NSA und zum US-Verteidigungsminsisterium hat, aktives Gründungsmitglied der KRA ist und als Urheber des Key Escrow Konzeptes gilt. TIS

stellt Firewallprogramme mit Key Recovery Features (Gauntlet) und ein eigenes Key Recovery System namens "RecoverKey" (das nach Aussage von Phil Zimmermann aber in keines der PGP Programme integriert wird) her. Ihr CEO, Stephen Walker, war jahrelang bei der NSA, der Defense Advanced Research Projects Agency (DARPA) und dem Office of the Secretary of Defense angestellt. Neben Walker rekrutieren sich viele der bei TIS Beschäftigten aus ehemaligen Angestellten und Kryptologen der NSA.
Am 12. November 1998 tritt NAI der KRA wieder bei. Es ist anzunehmen, dass dieser Schritt in engem Zusammenhang mit der Aquisition von TIS steht.
In einer E-Mail vom 13.11.1998 an die amerikanische PGP-User Mailingliste (Message-ID: (19981113233032.21663.00000701@ng105.aol.com in alt.security.pgp) bemerkt Will Price, Architect/Senior Manager für PGP Client Produkte der Total Network Security Abteilung, dass die Aufführung von NAI auf der KRA Mitgliederpage einzig das Resultat des Aufkaufs von TIS sei. NAI habe nach der TIS Aquisition ein Paket von Produkten mit eingebauten Key Escrow Funktionen erworben, deren Entfernung oder Modifizierung im Kontext von Überlegungen zu Export und Key Escrow, so dass sie weniger Big Brother haft arbeiten, viel Zeit in Anspruch nehmen wird. Alle Punkte hätten keine Auswirkungen auf die PGP Gruppe und NAI würde weite wie bisher damit fortfahren, den vollen Sourcecode zu publizieren.
Zum Thema "Back-Doors" in PGP 5/6 veröffentliche ich an dieser Stelle noch eine E-Mail Benachrichtigung von Phil Zimmermann vom 03.06.1999:

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

I'd like to address the rumors concerning the cryptographic integrity of PGP, including recent versions made by Network Associates, as well as recent freeware versions built and released by Stale Schumacher on his website in Norway at http://www.pgpi.org. These rumors allege that these versions of PGP contain back doors for the US Government to access the plaintext messages or keys. I do not know how such sensationalist conspiracy theories get started, but they seem to come from people who believe that The X-Files is a documentary.

Let me assure everyone that all versions of PGP that are released from Network Associates have the same cryptographic integrity as all previous versions of PGP that were released since the old days before I started my company, PGP Inc. In fact, no version of PGP in which I have been personally involved has ever had any back doors or any other mechanism to intentionally weaken PGP. That includes versions released by MIT, PGP Inc, Network Associates, or Stale Schumacher.

After all the hardship and legal persecution that I endured to bring PGP to the world, I find it surprising and offensive that anyone would think that I would quietly stand by and tolerate any compromise in the cryptographic interity of PGP.

When Network Associates acquired my company in December 1997, they also acquired the same engineering team that we had put together at PGP Inc, a team dedicated to the same principles of personal privacy that led me to create PGP. This team is still working on PGP today, and will continue to help me protect the integrity of PGP. Network Associates has not shown the slightest interest in compromising the integrity of PGP. They recognise that it would not be in their business interests to do so.

We have always published the source code for every version of PGP for peer review purposes, and Network Associates has carried on that tradition. Anyone may download the source code for PGP from www.pgpi.org and examine it for any back doors. Stale Schumacher, an independent PGP activist who is not an employee of Network Associates, has done all the builds since PGP 5.0i for the freeware versions of PGP in Europe. I have known Stale for several years and I know that he is committed to the same political principles of privacy as I am. I feel confident that Stale would never compromise the integrity of PGP in the versions that he builds for distribution on his site. Nonetheless, anyone who worries if the binary executables for PGP are trustworthy may compile the code themselves and rebuild the binaries for their own personal use, as long as they do not redistribute such rebuilt binaries for others to use.

-- Philip Zimmermann
http://www.pgp.com/phil
3 June 1999

-----BEGIN PGP SIGNATURE-----
Version: PGP 6.5.1b40

iQA/AwUBN1bDM2PLaR3669X8EQLXSACg4Z5+//BgNg4OjeKDugnQ0wmWbXEAoPsl
v4z0is5aXeLPf0cOJSqnyX9Q
=QOqg
-----END PGP SIGNATURE-----

Links:

- WIRED: Pretty Good Privacy Not Looking So Great
- WIRED: Another Network Associates U-Turn on Key Recovery ?
- WIRED: NAI Back in Key Recovery Group
- WIRED: Network Associates Disavows Key Recovery Tie

Installation oder
"Ich habe das Programmpaket, was nun ?"

Installation

Klicke 2x mit der Maus auf die Archivdatei im Explorer, danach startet automatisch die Installation von PGP 5/6

Zuerst startet das Auswahlfenster zur Selektion der zu installierenden PGP Komponenten wie das PGP Hauptprogramm, PGPdisk, PGPnet und diverse E-Mail Plug-In Module. Ab Version 6.5.2 a erkennt PGP automatisch, welche E-Mail Plug-Ins zu installieren sind und selektiert diese auch automatisch.

Noch zwei Bemerkungen zu PGPdisk und PGPnet.

Bei PGPdisk handelt es um ein Programm zur Erstellung von Containerdateien, in denen weitere Dateien abgelegt werden können. Wird eine Containerdatei geöffnet, so wird sie als zusätzliches Laufwerk im Explorer eingebunden. Schliesst man die Containerdatei, werden die darin enthaltenen Dateien mit der Containerdatei mittels CAST verschlüsselt und das zusätzliche Laufwerk verschwindet im Explorer
Es kann nicht schaden, PGPdisk zu installieren, aber nur ab den PGP Versionen 6.0.2, da PGPdisk in den vorangegangenen Versionen Sicherheitsfehler besitzt. Zur Vorabinformation, ob man PGPdisk nutzen möchte siehe PGPdisk oder "Wie verwahre ich Daten auf meiner Festplatte?"
Bei PGPnet handelt es sich um ein Programm zum Aufbau von verschlüsselten Verbindungen von einem Rechner zu einem anderen Rechner oder Netzwerk, den Virtuel Private Networks ähnlich und basierend auf dem IPsec Standard. Die Konfiguration und Anwendung ist ziemlich kompliziert.
Deshalb sollten nur diejenigen, die sich damit auskennen und verschlüsselte "Live"-Verbindungen benötigen, PGPnet installieren. Ausserdem kann es zu Problemen mit dem Netzwerk kommen, wenn PGPnet installiert wird. Ein installiertes PGPnet kann nur durch Deinstallation des gesammten PGP Programmes wieder vom Rechner entfernt werden. Zur Vorabinformation siehe PGP Virtuelles Privates Netzwerk PGPnet und PGPnet oder "Die Konfiguration und Anwendung des VPN PGPnet"

Nach der Installation befinden sich im Startmenü mehrere Einträge:

PGPtray

legt PGP als Trayicon in der unteren rechten Ecke der Startleiste ab.
Über einen linken oder rechten Mausklick auf das PGPtray Icon öffnet sich das Kommandomenü in dem folgende Funktionen aufgerufen werden:

- Optionen einstellen (PGP Preferences, bzw. Options)
- PGPkeys zur Schlüsselverwaltung aufrufen
- Schnellstartbuttonleiste (PGPtools)
- PGPdisk zur Verwaltung verschlüsselter Containerdateien (ab PGP 6.0 for Business Security)
- Verschlüsseln, Signieren, Entschlüsseln und Überprüfen des Clipboardinhaltes

PGPkeys

Über PGPkeys öffnet sich ein Fenster, das den gesamten Inhalt des Pubrings anzeigt und in dem alle Kommandos zur Schlüsselverwaltung ausgeführt werden können, d.h. darüber wird auf den Public Key und Secret Key Ring zugegriffen und Keys erzeugt, exportiert, signiert, ausgeschaltet, versendet und geholt. Die Veränderung der Anzeige erreicht man durch Aus-

wahl des Menüpunktes "Keys "und dann "Select Columns", bei PGP 6 über "View" und der direkten Auswahl der Spalten. Dabei bestimmt die Reihenfolge, in der die Anzeigeoptionen aktiviert werden auch die Reihenfolge (von links nach rechts) ihrer Anzeige im PGPkeys Fenster. Will man also als erste Spalte *ADK* angezeigt bekommen, muss *ADK* auch als **erste** Anzeigeoption aktiviert werden.

Die zu verwendenden Anzeigespalten:

ADK	handelt es sich um einen CMRK-Public Key mit Drittkeyverschlüsselung, wird dies durch einen roten Warnknopf angezeigt (Wichtig !), Keys ohne Vertrauens- und Authentizitätsparameter erscheinen mit grauem Knopf, mit diesen Parametern versehen mit einem grünen Knopf
Creation Date	Datum der Schlüsselerzeugung
Expiration Date	Gültigkeitsdauer des Keys
Description	Typ (RSA/DH/DSS), Art (Public Key, Signatur, User-ID), Status (exportierbar, ausser Funktion)
Key ID	"Identifikationsnummer" des Keys
Size	Keylänge in Bits
Trust	Grad der Vertrauensstufe
Validity	Grad der Gültigkeit/Authentizität

PGPadmin

mit dem zu den Business Security Versionen von PGP 5.5 und 5 gehörenden PGPadmin kann eine Einzelbenutzerversion (PGP Clientversion) erstellt und bis ins Detail vorkonfiguriert werden. Detailierte Informationen zu den Konfigurations- und Einschränkungsmöglichkeiten finden sich im Kapitel Abschaffung der Privacy bei PGP 5/6 und die Diskussion um GAK, CMRK, ARR, MRK.

PGPtools

PGPtools stellt eine Schnellstartleiste zur Verfügung über die

- PGPkeys aufgerufen
- eine Datei verschlüsselt (encrypt) & entschlüsselt (decrypt)
- eine Datei signiert (sign) & und überprüft (verify)
- eine Datei verschlüsselt & signiert
- eine Datei gelöscht (wipe)
- der freie Speicherplatz einer Partition oder Festplatte mit Freespace Wipe gelöscht und überschrieben

wird

PGPdisk

mittels PGPdisk kann eine durch CAST/SHA-1 verschlüsselte und mit einer Passphrase gesicherte Containerdatei ("Dateiname.pgd") erstellt werden, die als virtuelles Laufwerk geladen (mount) oder entladen (unmount) wird und als solches im geladenen Zustand für alle Dateioperationen genutzt werden kann. Wird die Containerdatei entladen, verschlüsselt PGPdisk die gespeicherten Inhalte der Containerdatei und das virtuelle Laufwerk verschwindet aus der Liste der verfügbaren Laufwerke. Die gleichen Funktionen sind auch direkt über ein Explorerkontextmenü verfügbar, wenn die Containerdatei im Explorer ausgewählt wird.

PGP Explorerkontextmenü

Im Explorerkontextmenü zu einer Datei (Aufruf über Markieren der Datei und 1xMKrT) findet sich der Eintrag "PGP", über den man direkt PGP-Operationen durchführen kann. Dabei verändert sich der Eintrag gemäss dem Dateityp.

Möglichkeiten

»Dateiname.asc« - eine Datei entschlüsseln und/oder Signatur überprüfen ("Decrypt/Verify")

```
PGP 2.6.3>  pgp  Datei.asc/Datei.pgp
```

»Dateiname.xyz« - die Datei verschlüsseln ("Encrypt"),

```
PGP 2.6.3> pgp -e Datei.xyz Empfänger-User-ID
```

signieren ("Sign")

```
PGP 2.6.3>  pgp -s Datei.xyz Eigene-User-ID
```

oder beides zusammen ("Encrypt and Sign")

```
PGP 2.6.3> pgp -es Datei.xyz Empfänger-User-ID Eigene-User-ID
```

»Pubring.pkr/Secring.skr« - dem Keyring einen Key hinzufügen ("Add Key")

```
PGP 2.6.3> pgp -ka Datei-mit-Key lw:\pfad\pubring.pgp
```

löschen und überschreiben ("Wipe")

über den Eintrag "PGPdisk" können die Funktionen zur Verwaltung einer PGPdisk Containerdatei ("Dateiname.pgd") aufgrufen werden:

- PGPdisk laden/entladen (Mount/Unmount PGPdisk)
- Passphrase hinzufügen (Add Passphrase), ändern (Change Passphrase) oder entfernen (Remove Passphrase)
- Public Key, für den die PGPdisk verschlüsselt wird hinzufügen/entfernen (Add/Remove Public Keys)

PGP einrichten oder
"Was und wie kann ich alles bei PGP einstellen ?"

Nach der Installation von PGP folgen nun die ersten Einstellungen

Optionen

Aufruf über 1xMK auf das PGPtrayicon, Menüpunkt "PGP Preferences"

GENERAL

Encryption and Signing Preferences

Always encrypt to default Key

bei 2.6.3 wird dazu in die Datei **config.txt** der Eintrag "**EncryptToSelf=on**" gesetzt, was bedeutet, dass eine Datei, Text o.a. nicht nur mit dem Public Key des Empfängers, sondern zusätzlich noch mit dem eigenen Public Key verschlüsselt wird.
Der Absender kann dann nachträglich, ebenso wie der Empfänger, die verschlüsselte E-Mail wieder entschlüsseln.

Beipiel

Eine verschlüsselte E-Mail wurde versendet. Tage später schreibt der Empfänger, bei der Übermittlung sei ein Fehler aufgetreten, er könne den Text nicht entschlüsseln. Da der Absender sehr viele E-Mails versendet, kann er sich nicht mehr erinnern, wer der Empfänger eigentlich ist, geschweige denn, um was es eigentlich ging. In dem Fall kann der Absender die aufbewahrte E-Mail aus dem Outboxfolder seines E-Mail Programmes heraussuchen und nach Eingabe seiner Passphrase wieder entschlüsseln, so dass der originale Text zur weiteren Bearbeitung wieder vorliegt.

- wenn sichergestellt werden kann, dass niemand den eigenen Secret Key erhalten kann, sollte man diese Funktion aktivieren.

Anmerkung

Wenn der Standard Public Key des Absenders ein DSS/DH-Key, der Empfängerkey aber ein RSA-Key ist, wird PGP bei einer Verschlüsselung die Warnung ausgeben, die besagt, dass der Empfänger, wenn er nicht PGP 5/6 benutzt, sondern eine 2.6.X Version, die E-Mail nicht entschlüsseln kann - und so ist es auch.

Cache...passphrase for...seconds

Bedeutet, dass nach der ersten Eingabe der Passphrase, diese bis zu 300 Sekunden lang im Arbeitsspeicher gehalten wird, so dass während dieses Zeitraums bei einer weiteren Entschlüsselung oder Signierung die Passphrase von PGP automatisch verwendet wird. Vorsicht: In dieser Zeitspanne sollte der PC nicht unbeaufsichtigt gelassen und PGP ordentlich beendet werden, damit die Passphrase nicht eventuell in der Swapdatei verbleibt oder ein Unbefugter die Entschlüsselung vornehmen kann.

- Sicherheitshalber sollte hier die Option eingeschaltet, die Zeit aber auf 1 Sekunde gesetzt werden, da ansonsten die Passphrase auf der Festplatte oder im Arbeitsspeicher rekonstruierbar ist.

Detaillierte Informationen zur Passphrase, auch "Mantra" genannt, finden sich in der Mantra FAQ

Show recipients when encrypting to marginally valid Keys

Bedeutet, dass PGP eine Warnmeldung ausgibt, wenn bei der Verschlüsselung ein Public Key benutzt wird, dessen Vertrauensgrad ("Trust") nur als geringfügig eingestuft ist. Es gibt: "nicht", "geringfügig" und "voll vertrauenswürdige" Public Keys. Normalerweise sind alle Public Keys solange als nicht vertrauenswürdig einzustufen, solange man sich nicht persönlich davon überzeugt hat, dass der Public Key wirklich dem User XY gehört.

- kann eingeschaltet werden, da informativ, allerdings sieht man den Trust/Validity Grad schon im Keyauswahlfenster.

Meines Wissens arbeitet diese Funktion nicht.

Key Generation Preferences

Faster Key generation

Betrifft die Erzeugung von DSS/Diffie-Hellmann Keys, deren Länge fest vorgegeben ist. Bei der Errechnung des Schlüsselpaars bei diesen Längen wird ein im voraus berechneter Satz von Primzahlen benutzt, um die Geschwindigkeit zu erhöhen.

- sollte abgeschaltet werden

File Wiping Preferences

Display wipe confirmation dialog/Warn before wiping

PGP besitzt eine eingebaute Dateilösch- und überschreibungsfunktion, wobei der Dateiname an sich in einem Verzeichnis immer noch rekonstruierbar bleibt. Werden im Explorer Shortcuts über das PGP Kontextmenü überschrieben, ist nur die Zieldatei betroffen, der Shortcut bleibt bestehen, nur bei Verwendung des Wipe-Befehls über die PGPtools werden beide gelöscht. Wird diese Option aktiviert, fordert PGP eine Bestätigung ein, bevor eine Datei überschrieben wird.

Number of passes

hier kann eingestellt werden, wie oft der Inhalt einer zu löschenden Datei überschrieben wird, möglich sind 1 bis 32 Durchgänge, PGP selbst empfiehlt 10 Durchgänge, da bekannt sei, dass Firmen, die sich mit der Wiederherstellung von Daten beschäftigen, auch Daten rekonstruieren könnten, die bis zu 9 x überschrieben wurden. Während des Löschens und Überschreibens freien Speicherplatzes (Free Space Wipe) müssen alle anderen Applikationen geschlossen werden, da der Free Space Wipeprozess durch jeden anderen Schreibprozess zurückgesetzt wird. Zum Thema File Wiping siehe auch Nachteile und Sicherheitsrisiken von PGP 5/6 gegenüber PGP 2.6.3

FILES

In diesem Menü werden die Pfade zum Pubring und Secring angegeben

Möglichkeiten

a. Beide liegen im PGP-Verzeichnis auf der Festplatte
   Am praktischsten, aber auch am unsichersten, da jederzeit auf den privaten Schlüsselbund zugegriffen werden kann.
b. Beide liegen auf einem Wechselmedium (z. B. Diskette)
   Am sichersten, nachteilig wirkt sich aus, dass ein Dateizugriff im Explorer parallel zu einem Zugriff auf das Floppylaufwerk führt, da PGP auch im Explorer integriert ist. Also muss die PGP-Diskette ständig im Laufwerk liegen.
c. Pubring auf Festplatte, Secring auf Wechselmedium
   Sicher, da der Secring auf dem Wechselmedium untergebracht ist, nur der Zugriff auf den Pubring ist gegeben.
   Bei Dateioperationen im Explorer findet ein Zugriff auf das Floppylaufwerk nur statt, wenn im Kontextmenü PGP angewählt wird.
d. Pubring und Secring liegen auf einer verschlüsselten Partition oder in einer verschlüsselten Containerdatei auf der Festplatte.
   Dazu besorgt man sich das Programm ScramDisk und das dazu passende Secure Tray Util, erstellt eine neue, leere Partition und formatiert sie mit ScramDisk um oder erzeugt eine einzelne, mindestens 1 MB grosse Containerdatei.
   Nachdem man alle Schlüsselringadteien in die Partition oder die Containerdatei verschoben hat, werden die Pfade zu den Schlüsselringdateien in den PGP Optionen angepasst.
   Der Nachteil dieser sicheren Methode besteht darin, dass man vor Benutzung von PGP die Partition oder die Containerdatei mounten muß.

Bei **PGP 2.6.3** werden dazu in der Datei **config.txt** die Enträge
PubRing = "LW:\pubring.pgp"
SecRing = "LW:\secring.pgp"
benutzt

EMAIL

Neben Optionen für alle E-Mail Programme stehen hier Optionen die nur E-Mail Programme betreffen, für die ein PGP Plug-In existiert, wie Qualcomm's EUDORA, Microsoft's OUTLOOK und EXCHANGE, Macintosh's CLARIS. Für OUTLOOK (EXPRESS) gibt es ein Plugin, in PGP 6 ist es schon enthalten. Eine anschauliche Einführung für EUDORA Benutzer gibt es bei Kryptographie und Datenschutz im Internet - Wie benutze ich PGP ? Für CLARIS müsste ein Macuser etwas schreiben, da ich nicht mit einem Mac arbeite.)

Use PGP/MIME when sending email

Betrifft E-Mail Programme, für die ein PGP 5.X Plug-In existiert und die die MIME Implementation von PGP unterstützen. Ist diese Option aktiviert, muss bei diesen Programmen die PGP/MIME Funktion nicht mehr aktiviert werden. In diesem Fall werden E-Mails automatisch ver- / entschlüsselt und signiert.

- Voraussetzung für einen sinnvollen Einsatz ist die Unterstützung des PGP/MIME Standards auf der Empfängerseite, die zur Zeit generell noch nicht gegeben ist.

Word wrap clear-signed messages at column...

Damit wird eingestellt, ab welcher Breite der Zeilenumbruch der PGP Signatur bei E-Mails erfolgen soll, die unverschlüsselt versendet, aber mit einer PGP-Signatur versehen werden. Am besten, man stellt die für Postings und E-Mails allgemein empfohlene Länge von "75" ein, oder gleicht die PGP Einstellung für den Zeilenumbruch mit dem Wert, den man im E-Mailprogramm verwendet, an. Manche E-Mailprogramme brechen den Text während des

Verfassens der Nachricht am rechten Rand des Textfensters um und nicht nach einer voreingestellten Länge. Der eigentliche Zeilenumbruch findet dann erst vor, bzw. bei der Versendung statt. Verschlüsselt oder signiert man diesen Text, kann durch den nachträglichen Umbruch die Signatur oder der gesamte verschlüsselte Text so zerstückelt werden, dass der Empfänger die message nicht entschlüsseln oder die Signatur nicht überprüfen kann.

Encrypt new mesages by default

alle E-Mails werden immer an den Empfänger verschlüsselt, so lange ein Public Key des Empfängers vorhanden ist.
Für E-Mailprogramme, die den PGP/MIME Standrad und die PGP Plug-Ins unterstützen.

Sign new messages by default

alle Postings und E-Mails werden immer signiert.
Für E-Mailprogramme, die den PGP/MIME Standrad und die PGP Plug-Ins unterstützen.

Automatically decrypt/verify when opening messages

Wird eine PGP verschlüsselte E-Mail geöffnet, wird sie, wenn gleichzeitig das caching der Passphrase eingestellt ist, automtaisch entschlüsselt, bzw. die enthaltene Signatur geprüft.
Für E-Mailprogramme, die den PGP/MIME Standrad und die PGP Plug-Ins unterstützen.

SERVER

Hier werden die Keyserver eingetragen, die benutzen werden, um

- Public Keys zu verbreiten
- geänderte Public Keys auf dem Keyserver abzugleichen
- Rückzugsurkunden kompromittierter Public Keys zu veröffentlichen
- Public Keys zu bekommen oder zu suchen
- Public Keys zu löschen (betrifft nur die Keys auf dem NAI eigenen Certification Server)

Da sich das Format der PGP 5/6 Keys von dem der PGP 2.6.X Versionen unterscheiden, muss man verschiedene Keyserver benutzen: Es gibt PGP 5/6 kompatible Keyserver, die über das WWW benutzt werden und am Port 11371 oder 389 "lauschen".
Nähere Informationen und Zugriffsmöglichkeiten erhält man über das PGP.NET.
Daneben kann gibt es auch Keyserver wie z. B. pgp-Public-Keys@informatik.uni-hamburg.de, die per E-Mail bedient werden. Eine E-Mail mit dem Subject: HELP ohne Body sendet Dir z. B. den Hilfetext des Keyservers zu. Diese Keyserver sind jedoch auf RSA-Keys spezialisiert und nehmen keine DSS/DH-Keys an.

PGP 6 bietet ein erweitertes Keyservermenü, über das einige Konfigurationseinstellungen vorgenommen werden können: Über den Button Edit wird ein Keyserver näher spezifiziert:

Server Information

Protocol:

- HTTP
  Keys werden über das Hypertext Transfer Protocol versendet.

- LDAP (Lightweight Directory Access Protocol)
  Keys können über ein spezielles Verzeichniszugriffs- und Suchprotokoll nach bestimmten Kriterien gesucht werden.
  Keys die versendet werden, werden in einer Verzeichnisstruktur gespeichert.
- LDAPS
  gleiches Protocol wie oben, zusätzlich wird die Kommunikation zum Keyserver über die PGP eigene SSL Implementation TLS (Transport Layer Security) verschlüsselt und authenifiziert.

  Über LDAPS kann man neben der Keysuche und -versand, nach Ausweisung über den eigene Key, auch eigene Keys selbst auf dem Keyserver deaktivieren oder löschen.
  Der PGP 6 kompatible Keyserver ldaps://certserver.pgp.com ist z. B. so konfiguriert.

Server name:

hier wird der Domainnamen oder die IP-Adresse des Keyserver eingetragen

Port:

- 11371 für HTTP-Keyserver
- 389 für LDAP(S)-Keyserver

Server Key:

Key eines LDAPS-Keyservers, der zur Authentifizierung einer Verbindung benutzt wird

Serves keys for Domain:

- Any Domain: Keys jeder Domain der Key E-Mail Adresse werden an diesen Keyserver gesendet
- zweites Feld: Nur Keys mit einer speziellen Domain der Key E-Mailadresse werden an diesen Keyserver versendet (z.B. bei Angabe somewhere.de werden nur Keys mit E-Mailadressen, die @somewhere.de enthalten, an den Keyserver versendet)

List in search window

bei Aktivierung der Checkbox wird der Server im Menü Server Send to und Search aufgeführt

Synchronize with server upon

Encrypting with unknown keys

soll an einen Empfänger eine Nachricht verschlüsselt werden, dessen Public Key sich nicht im Pubring befindet, wird der Public key vom Keyserver angefordert (wie immer das auch funktionieren soll, wenn sowieso nur Empfänger im Keyauswahlfenster ausgewählt werden können, deren Key auch im Pubring vorhanden sind).

Signing keys

wenn ein Key signiert werden soll, wird zuerst der Keyserver nach der aktuellsten Version des Public Keys abgefragt, bzw. überprüft, ob der betreffende Public Key nicht in der Zwischenzeit zurückgezogen wurde. Nach Download des aktuellen Keys und Update des Pubrings, wird dieser, mit der Signatur versehen, an den Keyserver gesendet.

Adding names/photos/revokers

bevor eine zusätzliche User-ID, ein Foto oder ein Designated Revoker einem Public Key hinzugefügt wird, wird die Aktualität des Public Keys mit dem Keyserver abgeglichen, um zu verhindern, dass die Zusätze nicht zu einem bereits zurückgezogenen Key erfolgen.

Revocation

bevor die Revocation eines Public Keys erzeugt werden kann, wird der Public Key erneut vom Keyserver angefordert, um zu verhindern, dass die Revocation für einen bereits zurückgezogenen Key ausgestellt wird.

Verification

wenn eine Datei/Nachricht entschlüsselt oder deren Signatur überprüft wird und es finden sich Signaturen von Public Keys, die im lokalen Pubring nicht vorhanden sind, werden die fehlenden Public Keys automatisch vom Keyserver angefordert.

ADVANCED

Encryption

im Feld "Enabled algorithms" werden die Verschlüsselungsalgorithmen CAST, Triple-DES und IDEA aktiviert, die benutzt werden sollen, um Dateien oder Texte konventionell zu verschlüsseln, oder die bei der Public Keyerzeugung und späteren Benutzung herangezogen werden können.
Welcher Algorithmus letztendlich benutzt wird, legt man über den "Preferred algorithm" (zu bevorzugenden Algorithmus) fest.
Bei der Keyerzeugung wird dem Public Key eine Zusatzinformation beigefügt, welchen Algorithmus das PGP Programm des späteren Absenders benutzen soll, wenn es den Public Key des Empfängers zur Verschlüsselung anwendet.

DH Keys können CAST, Triple-DES oder den IDEA Algorithmus enthalten
RSA Keys können nur mit dem IDEA Algorithmus arbeiten.

Aus diesem Grund ist es **wichtig**, wenn man einen RSA Key erzeugen will, vorher IDEA als bevorzugten Algorithmus anzugeben. Auch aus Kompatibilitätsgründen ist es angebracht, zwar alle Algorithmen zu aktivieren, als bevorzugten Algorithmus aber stets IDEA einzutragen.

Kurzinfos zu den verwendeten symmetrischen Algorithmen

Alle drei Algorithmen operieren mit 64-bit Datenblöcken und PGP verwendet sie im 64-bit cipher feedback (CFB) Modus.

CAST

wurde nach seinen Erfindern Carlisle Adams und Stafford Tavares, die ihn für die Northern Telecom (Nortel) entwickelt haben, benannt.
CAST ist ein freier, schnell arbeitender A. mit einer Keygrösse von 128-bit, der noch nicht so lang und gut untersucht ist wie IDEA.

Triple-DES

Der freie DES (Data Encryption Standard) wurde von IBM unter dem Namen Lucifer, der NSA und dem NIST in den 70er Jahren entwickelt und hat eine Keygrösse von 56-bit.
Bei dem seit einigen Jahren untersuchten Triple-DES Algorithmus werden drei unterschiedliche Keys dreimal auf einen Datenblock angewendet, so dass sich eine Gesamtkeygrösse von 168-bit ergibt, die effektive Keystärke, bezogen auf einen Angriff soll bei 112-bit liegen.

IDEA

Der International Data Encryption Algorithm (IDEA) wurde von James L. Massey und Xuejia an der ETH in Zürich entwickelt und 1990 veröffentlicht und von PGP seit dem Anfang verwendet.
IDEA ist gegenüber der differentiellen und linearen Kryptoanalyse mehr resistent als DES.
Die Keygrösse liegt bei 128-bit, die Lizenz an IDEA bei der Firma Ascom Systec.

Trust Model

Display Marginal Validity Level

Über diese Option wird unter der Spalte "Validity" (Gültigkeit) der Grad der Authentizität, die man einem Key zubilligt, entweder in Form verschiedenfarbiger Knöpfe/Rauten oder verschiedenschraffierter Balken angezeigt.

Treat Marginally Valid Keys as Untrusted

ausser Funktion

Warn When Encrypting Keys to keys with an ADK

ausser Funktion, man sieht nur, dass der ADK-Key zusätzlich im Keyauswahlfenster erscheint und mit einem kleinen, gelben Schlosssymbol versehen ist.

Export Format

Compatible

bei dem Export eines Public Keys aus dem Pubring (und speichern als Datei) wird ein Dateiformat verwendet, das zu den vorherigen PGP Versionen kompatibel ist.
Diese Option sollte immer aktiviert sein mit der Ausnahme, dass bekannt ist, dass der Empfänger ebenfalls PGP 6 benutzt.

Complete

bei dem Export eines Public Keys aus dem Pubring (und speichern als Datei) wird das PGP 6 Dateiformat verwendet, das auch Angaben zu Foto-IDs und Designated Revokers enthält und nicht kompatibel zu vorherigen PGP Versionen ist.

Schlüssel oder
"Wie komme ich an meinen Secret und Public Key ?"

Allgemeine Keybestandteile

Was so alles zu einem Key gehört, erfährt man, wenn PGPkeys geöffnet wird (mindestens die Keys von Philip Zimmermann, dem Entwickler von PGP müssten vorhanden sein).
Als erstes sieht man ein Fenster mit Schlüsselabbildungen in zwei verschiedenen Farben. Gelbe Keys stehen für DSS/DH Keys, blaue oder graue Schlüssel für RSA-Keys. Eine paarweise Abbildung besagt, dass zum Public Key ein zugehöriger Secret Key vorliegt, also sind das die eigenen Schlüssel. Dahinter steht die User-ID und die E-Mailadresse des Besitzers.
Als weitere Spalten können folgende Anzeigen über "View" ("Select Columns") aktiviert werden (die Reihenfolge der Aktivierung bestimmt die Reihenfolge der Anzeige):

- **Validity** (Gültigkeit)
  je nach Grad als graue, dunkelgraue, schraffierte oder farbige Balken oder Buttons
- **Size** (Keylänge)
  als Byteangabe
- **Description** (Beschreibung des Key und Signaturtyps)
  RSA oder DH/DSS Public Key oder Key Paar, Revoked Public Key, einfache, nur im lokalen Pubring gültige Signatur (RSA,DH/DSS signature) oder mitzuversendende Signatur (RSA,DH/DSS signature exportable)
- **Key ID** (Key Identifikationsnummer)
  in hexadezimaler Angabe "0xYYYYYYYY"
- **Trust** (Vertrauen)
  je nach Grad als grauer, dunkelgrauer oder schraffierter Balken
- **ADK** (Additional Decryption Key)
  als grauer (kein ADK-Key) oder farbiger (ADK-Key) Button
- **Creation** (Erstellung)
  als Datum der Keyerstellung
- **Expiration** (Verfall)
  als Datum, an dem der Key ungültig wird

Wenn ein Key markiert wird und der Menüpunkt "Keys", "Keys Properties" angewählt wird, erscheint ein neues Fenster, in dem die obigen Angaben nochmals aufgeführt sind plus der folgenden, zusätzlichen Eigenschaftsangaben:

- die Gültigkeit unter "Validity" in zwei Zustandsformen
- das Vetrauen unter "Trust" mit einem Schieberegler, über den der Grad des Vetrauens in drei Stufen eingestellt wird
- unter "Cipher" der Verschlüsselungsalgorithmus (CAST, IDEA oder Triple-DES), der vom Mailpartner verwendet wird, wenn mit dem Public Key verschlüsselt wird
- der Fingerprint des Keys
- in der Zeile "Additional Decryption Key" (ADK/CMRK), der Drittkey, an den bei CMRK-Keys mitverschlüsselt wird, bei PGP 6 befindet sich diese Angabe unter dem Karteireiter "ADK"
- die Foto User-ID bei PGP 6 DH/DSS Keys
- der Karteireiter "Subkeys" bei PGP 6 DH/DSS Keys, unter dem alle DH Encryption Keys aufgeführt sind
  Anmerkung:
  Auch PGP 2.6.3 in bietet das Feature, den Signier-Key von dem Verschlüsselung-Key durch die Angaben SIGN und ENCR zu unterscheiden
- der Button "Change Passphrase", über den nach Eingabe der alten Passphrase eine neue Passphrase für den Key vergeben werden kann

Bei PGP 2.6.3 kann man sich mit folgendem Kommando die Bestandteile eines Public Key anzeigen lassen:

```
PGP 2.6.3> pgp -kvc User-ID
```

Eine genauere Erklärung dieser einzelnen Punkte würde an dieser Stelle zu weit führen und unverständlich sein.
Alle Bestandteile werden im Laufe dieser Anleitung in ihrem Zusammenhang erläutert.

Schlüsselerzeugung

Vor der Keyerzeugung sollte man unbedingt noch einen Blick auf den Punkt ADVANCED werfen und sich die Mantra FAQ durchlesen.

1. 1xMK auf PGPtrayicon ", dann Launch PGPkeys" oder direkt im Startmenü auf das PGPkeyicon klicken
2. Menüpunkt "Keys"
3. "New Key"
4. Full Name: (Vorname) Nachname eingeben
5. E-Mail Address: user@adresse eingeben
6. Key-Typ wählen

   Anmerkung
   Es ist vorteilhaft, nacheinander sowohl ein RSA-Keypaar, als auch ein DH-Keypaar zu erzeugen, damit auch Benutzer anderer 2.6.X Versionen von PGP mit dem Keybesitzer kommunizieren können.

7. **Key Pair Size** (Schlüssellänge) wählen

Empfehlungen zur Schlüssellänge

Abseits der Diskussionen, welche Länge ausreicht oder nicht, sollte man m. M. nach eine Länge von mindestens 3100 bis 4096-bit in Verbindung mit einer starken Passphrase wählen. Dabei ist jedoch zu beachten, dass eine Keybenutzung mit PGP 2.6.3 (i/a) nicht mehr möglich ist, sondern nur bei Verwendung der PGP 2.6.3 IN - Version. PGP 2.6.3 i/a sollte nicht mehr verwendet werden.

Im Text "Probleme beim PGP-Einsatz in Zertifizierungsstellen und deren Lösung durch PGP2.6.3in und OpenPGP" von Ingmar Camphausen und Lutz Donnerhacke findet sich unter dem Kapitel "Lange Schlüssel":

"...Nach aktuellen Studien entsprechen sich die Schlüssellängen bei asymmetrischer und symmetrischer Verschlüsselung gemäss Tabelle, was ihre Resistenz gegen Angriffe angeht. Damit ist klar, dass die Entscheidung, in den IN-Zertifizierungsrichtlinien eine Schlüssellänge von 2048 bit RSA für die Root-CA und mindestens 1024 bit RSA für Nutzer zu fordern, nicht überzogen war.
PGP 5 (mit DH-Keys,Anm. d. Autors) kann diese Forderungen nicht erfüllen, da sein Signaturalgorithmus DSA entwurfsbedingt auf 1024 bit DLP (DLP: Discrete logarithm problem) beschränkt ist.
Das im Gegensatz dazu skalierbare ElGamal-Verfahren wurde in PGP 5 so implementiert, dass Unterschriften nach diesem Verfahren einer Veröffentlichung des geheimen Schlüssels gleich kämen. Hier muss also eine vernünftige OpenPGP Implementation abgewartet werden.

Gleichzeitig wird aus der Tabelle auch klar, dass eigentlich um 2500 bit RSA oder 3100 bit DLP der Normalfall seien sollten."

RSA-Modulus	symmetrisch	DLP-Modulus	symmetrisch
512	63		
786	76	1024	56
1024	86	**2048**	**112**
2048	**117**	**3072**	**128**
2500	**128**	4096	168
3072	139		
4096	157		

Bitlängen gleicher Kryptoresistenz

Empfohlene Keylängen, die im Jahr X noch als sicher angesehen werden können, gemessen an Moore's Law (das besagt, dass sich die Rechenkapazität eines Prozessors durch neue Typen des gleichen Prozessors alle 18 Monate verdoppelt), der Berechenungszeit mit einem 450 Mhz Pentium II Prozessor, bzw. der Berechnungszeit in Mips Jahren und der Höhe des zur Verfügung stehenden Bugets auf Seiten des Angreifers (das sich alle 10 Jahre verdoppelt) nach Lenstra und Verheul.
Beide Autoren sagen auch, dass durch eine zukünftige Weiterentwicklung in der Computertechnik, wie Quantum Computer, die bis jetzt nur in hypothetischer Form existieren, das Modell hinfällig werden könnte.
Eigene Berechnungen nach diesem Modell können hier durchgeführt werden.

Jahr	Länge sym. Key (Bits) (IDEA,CAST)	Länge asym. Key (Bits) (RSA,ElGamal,DH)	Subgroup DLP Keylänge (DSA,DSS)
2000	70	952	125
2005	74	1149	131
2010	78	1369	138
2015	82	1613	145
2020	86	1881	151
2025	89	2174	158
2030	93	2493	165
2035	97	2840	172
2040	101	3214	179

Nach diesem Modell wäre ein 1024-Bit RSA Key bis Mitte 2001, ein 2048-Bit RSA Key bis Mitte 2022 und eine DSS Signatur bis Mitte 2001 als sicher vor Berechnung anzusehen.
Also müsste ein Public Key mindestens eine Länge von 2048-Bit aufweisen, eine DSS Signatur ebenfalls. Nach den Angaben beider Tabellen ist die DSS Signatur mit 1024-Bit DSA von PGP 5/6 als nicht sicher einzustufen.

In den Frequently Asked Questions About Today's Cryptography? Version 4.0 der RSA Labors werden folgende Empfehlungen ausgesprochen:

Grad	Block Cipher	RSA	DSA/DSS
nach US-Exportgesetz	40 Bits	512-Bit	512/80 Bits
persönlicher Bereich (E-Mail, unwichtige Daten)	56/64 Bits	768 Bits	768/136 Bits
kommerzieller Bereich	128 Bits	1024 Bits	1024/160 Bits
militärischer Bereich	160 Bits	2048 Bits	2048/200 Bits

Da 56-Bit DES bereits gebrochen wurde, sind die obigen Angaben veraltet und der minimalste Grad beim kommerziellen Bereich anzusetzen. Nimmt man die Angaben der beiden anderen Tabellen hinzu, wird klar, das der Mindeststandard im militärischen Bereich zu finden ist.
Ein Public Key muss demnach mindestens eine Länge von 2048-Bits aufweisen.

8. **Key Expiration**
hier kann man angeben, ob die Keys unbegrenzt oder zeitlich begrenzt genutzt werden sollen.
Never: Keys sind unbegrenzt benutzbar
Expires in xy Tagen: Keys sind nur xy Tage benutzbar.

Anmerkung
Das bedeutet für den Keybesitzer, dass er nach Verfall des Keys weiterhin an ihn verschlüsselte E-Mails entschlüsseln, aber keine neuen E-Mails mit diesem Key signieren kann. Für den E-Mailpartner, der den Public Key benutzt, heisst das, er kann nach Verfall des Keys keine E-Mails mehr an den Keybesitzer mit diesem Key verschlüsseln, aber noch dessen Signaturen überprüfen. Das bedeutet auch, dass nach Verfall neue Keys erzeugt werden müssen.

9. **Passphrase**

"Hide typing" Wenn die Gefahr besteht, beobachtet zu werden, wird damit die Monitoranzeige der Passphrase unterdrückt.

Anmerkung
Wähle eine möglichst lange und aus zufällig gwählten Zeichen bestehende Passphrase, um das Erraten oder Berechnen der Passphrase zu verhindern. Eine Passphrase die aus "Quark" besteht, wird leicht zu erraten sein, eine Passphrase, die aus "WerQu23 Das///?Nureiella51?? wekshh_/" besteht, dagegen nicht. Allerdings sollte man die Passphrase auch jederzeit nutzen können, denn wenn die Passphrase verloren geht oder vergessen wird, kann ein Key nicht mehr benutzt oder zurückgezogen werden!
Selbstverständlich ist dafür zu sorgen, dass nur der Keybesitzer allein Zugriff auf die Passphrase erhalten kann. Detaillierte Informationen zur Gestaltung einer sicheren Passphrase und einigen Hintergrundinformationen finden sich in der Mantra FAQ.
Nach Beendigung dieser Prozedur befindet sich der Public Key im Pubring. Dabei wurde nicht nur der Public Key an sich erstellt, sondern gleichzeitig dieser Public Key mit dem Secret Key unterschrieben. d.h. der Keybesitzer selbst hat seinen Public Key mit seiner eigenen Signatur versehen. Warum dieser Schritt automatisch

vollzogen wird und die eigene Signierung des Public Keys notwendig ist, beschreibt anschaulich die Eigensignatur (Selfsign) FAQ.

Befehl zur Schlüsselerzeugung bei **PGP 2.6.3**:

```
PGP 2.6.3> pgp -kg
```

10. **Key Revocation** erzeugen
    Die Key Revocation dient dazu, den Public Key als zurückgezogen, bzw. ungültig zu erklären.

    Anlässe, zu denen es nötig wird, die Key Revocation an einen Keyserver zu senden und die Kommunikationspartner davon zu unterrichten, sind zum Beispiel:

    - die versehentliche Löschung des Secret Keys
    - der Diebstahl des Secret Keys und der Passphrase

    Warum jetzt schon ?
    Es ist sinnvoll, bereits nach der Keyerzeugung die Key Revocation für den eigenen RSA- und DSS/DH-Public Key zu erzeugen, denn im Falle eines Verlustes oder Diebstahls des Secret Keys und der Passphrase, ohne vorher die Key Revocation erzeugt zu haben, kann der Keybesitzer die Key Revocation nicht mehr herstellen und damit seinen Key nicht als zurückgezogen, bzw. ungültig auf den Keyservern markieren.
    Eine Löschung des eigenen Public Keys durch den Administrator des Keyservers ist nicht möglich.
    Auch mit einem zurückgezogenen Key ist es weiterhin möglich, mit diesem Key verschlüsselte oder signierte Daten zu entschlüsseln, bzw. zu überprüfen.
    Es ist nicht möglich, mit einem zurückgezogenen Key Daten weiter zu signieren, bzw. mit diesem Key Daten zu verschlüsseln.

    ### Vorgehensweise

    1. Kopien des Public und Secret Keys herstellen (zum Beispiel durch Kopieren auf Diskette)
    2. PGPkeys aufrufen
    3. den eigenen Public Key markieren
    4. 1xMTrT, Eintrag "Revoke" oder Menü "Keys", Eintrag "Revoke" anklicken
    5. Sicherheitsabfrage mit "ja" bestätigen
    6. Passphrase eingeben, danach erscheint der eigene Public Key mit einem roten Querbalken im PGPkeys Window
    7. den Key exportieren (z. B. als "RSArevoke.asc" und "DHrevoke.asc") und sicher vor unbefugtem Zugriff abspeichern
    8. die Kopien von Public und Secret Key wieder zurückspielen

    Befehl zur Erzeugung einer Key-Revocation bei **PGP 2.6.3**:

    ```
    PGP 2.6.3> pgp -kd User-ID des eigenen Keys
    ```

Versendung einer Revocation

RSA-Key

Sende eine E-Mail an einen PGP Keyserver, z. B.
pgp-public-keys@informatik.uni.hamburg.de mit dem Subject: ADD.
In den Body wird nur die Datei "RSArevoke.asc" eingefügt, die ja bereits vorliegt.

DSS/DH-Key

Suche einen der PGP 5 kompatiblen WWW-Keyserver auf, wähle die Seite aus, wo Keys an den Server versendet werden können, und füge in das Formularfenster die Datei "DHrevoke.asc" ein. Eine andere Möglichkeit:
Der Keybesitzer ist noch im Besitz von Secret Key und Passphrase. Dann erstellt er die Revocation wie oben beschrieben und sendet sie direkt mit PGP an den Keyserver

Vorschlag

Auf die gleiche Weise kann man direkt die Public Keys als Datei "MeinPublicKey.asc" abspeichern, so steht der Public Key immer direkt zum Versand an E-Mail Partner bereit.

Verschlüsselung oder
"Erstellung und Versendung verschlüsselter E-Mails ?"

Verschlüsselt wird immer mit dem Public Key des EMPFÄNGERS, den der Empfänger zum Beispiel per E-Mail zugesendet hat und der sich nach dem Import mittels PGPkeys jetzt im eigenen Pubring befindet.

Verschlüsselung eines Textes via Clipboard

a) an Person A, der im E-Mailprogramm erstellt wurde

1. markieren und kopieren des Textes in die Zwischenablage
   oder man aktiviert im PGPtray Kontextmenü "Use Current Window" und markiert nur den gesamten Text
2. 1x Mausklick auf PGPtrayicon
3. 1x Mausklick auf Menüpunkt "Encrypt Clipboard", das Fenster "PGP - Selection Dialog (Schlüsselauswahl)" wird geöffnet
4. aus der Liste den Public Key von Person A auswählen, d.h. 1x Mausklick auf Eintrag, Maustaste gedrückt halten und den Eintrag in das untere Fenster "Recipients (Empfänger)" ziehen.
   Soll an mehrere Personen gleichzeitig verschlüsselt werden, weitere Public Keys der Personen B, C usw. in das untere Fenster ziehen.
5. soll ohne Public Key, aber konventionell über Vergabe einer Passphrase verschlüsselt werden, die Checkbox "Conventional Encryption" aktivieren
6. O.K.
7. im E-Mailprogramm den Text löschen oder markiert lassen, Funktion "Paste (Einfügen)" wählen. Darufhin wird der verschlüsselte Text eingefügt
8. Absenden

b) an eine Gruppe von Empfängern

1. in PGPkeys wird über das Menü *Groups/New Group* ein Gruppennamen für die Empfänger definiert
2. wenn man im Menü *Groups* den Menüeintrag *Show Groups* aktiviert hat, wird die neu erzeugte Gruppe im Gruppenfenster unterhalb des Keyfensters angezeigt
3. aus dem Keysfenster zieht man die Keys aller Gruppenmitglieder auf das Gruppenicon im Gruppenfenster, um sie der Empfängergruppe hinzuzufügen
4. im E-Mailprogramm erstellt man eine Verteilerliste mit der jeweiligen Option zur Erstellung von Mailing- oder Verteilerlisten, die alle E-Mailadressen der PGP Gruppenmitglieder beinhaltet
5. zur Verschlüsselung und/oder Signierung geht man wie unter a) vor, nur das anstelle des Keys einer Einzelperson der Gruppenname in das Empfängerfenster gezogen wird

Signierung oder
"Wie setzte ich eine digitale Unterschrift unter meinen Text ?"

Die digitale Signatur, die mit PGP erzeugt werden kann, dient dazu, bei einem Text auszuweisen, dass man wirklich der Verfasser ist.
Auf der Empfängerseite kann derjenige, der den Public Key besitzt, genauso die Authentizität des Textes kontrollieren.

Signierung via Clipboard

1. der gesamte Text wird markiert und in die Zwischenablage kopiert ("Kopieren/Copy") oder man aktiviert im PGPtray Kontextmenü "Use Current Window" und markiert nur den gesamten Text
2. 1xMK auf PGPtray
3. "Sign Clipboard" wählen
4. Auswahl des Signing Keys über das Drop-Down-Menü
5. Eingabe der Passphrase
6. den zuvor markierten Text mit dem Inhalt der Zwischenablage überschreiben ("Einfügen/Paste")

Hinweis zur Umlautcodierung

Es ist wichtig, dass die verfasste und signierte E-Mail vor oder bei der Versendung nicht mehr verändert wird.
Verändert sich nach der Signierung nur ein Zeichen, wird die Signatur ungültig.
Dies kann passieren, wenn im Body normale Umlaute verwendet wurden und die Umlaute der E-Mail vor der
Versendung durch die Verwendung des »Quoted-Printable«-Formats in 7-bit Zeichen oder durch ein Script,
das einen Umlaut wie "ä" in "ae" verändert, umgewandelt werden.

Hinweis zu PGP 6.X Cyber Knight Templar

Benutzt man mit der CKT PGP Version den SHA-1 oder RIPEMD 160 Algorithmus zur Signierung, kann eine verschlüsselte Datei von den PGP 2.6.3 Versionen **nicht** mehr überprüft, bzw. entschlüsselt werden.

Bei PGP 2.6.3 Empfängern ist es deshalb notwendig, als Hashalgorithmus MD5 in den Preferences von PGP 6.X CKT festzulegen.

Entschlüsselung, oder
"Wie gelange ich an den Ursprungstext ?"

Entschlüsselung einer vorliegenden Datei

Vorgehensweise im Explorer

1. Datei markieren
2. mit 1MKrT Kontextmenü aufrufen, darin "PGP" anwählen
3. "Decrypt/Verify" (Entschlüsseln und/oder Signatur prüfen) anklicken
4. im folgenden Fenster die Passphrase eingeben
5. Abspeichern der entschlüsselten Datei
6. bei (zusätzlicher) Signatur wird diese überprüft und
   ihr Status ("Good/Bad signature") ausgegeben

Entschlüsselung einer E-Mail oder eines verschlüsselten Textes innerhalb einer Datei über Clipboard

1. den gesamten PGP-Block,
   von ---BEGIN PGP bis einschliesslich ---END PGP,
   markieren
2. Markierten Teil ins Clipboard kopieren
3. Kontextmenü von PGPtrays mit 1MKrT aufrufen
4. Eintrag "Decrypt/Verify Clipboard" wählen
5. Passphrase eingeben
6. PGP gibt den Status der Signatur aus
7. Anzeige des entschlüsselten Textes über einen
   externen Editor (z. B. Notepad) oder
   den PGP eigenen Clipboardviewer

Entschlüsselung mittels PGP Plug-In (Beispiel Outlook)

- im Menü "Extras/Optionen/PGP kann durch Aktivierung der Option "Decrypt/Verify messages automatically" eingestellt werden, dass bei Eingang/Auswahl einer verschlüsselten mail PGP sofort mit der Entschlüsselung beginnt, d. h. automatisch der Signaturstatus ausgegeben und/oder die Eingabe der Passphrase abgefragt wird
- Ist obige Option deaktiviert, muss auf den linken Button mit dem geöffneten Briefumschlag geklickt werden, um die Signaturprüfung und/oder Entschlüsselung zu starten

Schlüsselverbreitung, oder
"Wie gelangen die Leute an meinen Public Key ?"

Um einen Public Key per E-Mail oder über eine Webpage zu verbreiten oder per E-Mail an einen Keyserver zu senden, muss der Public Key zunächst aus dem Pubring (Schlüsselbund) herausgelöst und als Datei abgelegt werden.

Der Export eines Public Keys

1. PGPkeys aufrufen
2. Public Key auswählen mit 1xMKlT auf den Eintrag des eigenen Keys (also "Name <user@E-Mailadresse>")
3. 1xMKrT Kontextmenü
4. Menüpunkt "Export"
5. Verzeichnis und Dateiname wählen, wobei wegen DOS nur 8 Zeichen für den Dateinamen gewählt werden sollten
6. soll auch der Secret Key exportiert werden, die Checkbox "Include Private Keys" aktivieren
   - **Vorsicht: der Secret Key darf unter keinen Umständen an Keyserver oder Mailpartner versendet werden**
7. bei Aktivierung der Checkbox "Include 6.0 Extensions" werden auch die zusätzlichen PGP 6 Formate zur Foto User-ID und Designated Revokers exportiert
   - deshalb nur aktivieren, wenn der Mailpartner ebenfalls PGP 6 benutzt oder der Key an einen PGP 6 kompatiblen Keyserver versendet wird
8. Public Key liegt als "MeinRSAPublicKey.asc" oder "MeinDSS/DHPublicKey.asc" vor

Export eines Public Keys bei **PGP 2.6.3**:

```
PGP 2.6.3> pgp -kxa Eigene-User-ID Dateiname LW:\pubring.pgp
```

Der Public Key kann und sollte veröffentlicht werden, damit jeder, der mit dem Keybesitzer eine verschlüsselte Kommunikation pflegen möchte, sich dessen Public Key besorgen kann. Um den Public Key zu veröffentlichen, gibt es mehrere Möglichkeiten:

Über Keyserver

auf den Keyservern liegen alle öffentlichen Schlüssel aller PGP-User weltweit, die ihren Public Key an diese Keyserver gesendet haben.

RSA-Keys

Sende eine E-Mail an einen PGP Keyserver, z. B. pgp-public-keys@informatik.uni.hamburg.de mit dem Subject: ADD
In den Body füge nur die Datei "MeinRSAPublicKey.asc" ein, die ja bereits vorliegt.

DSS/DH-Keys

Suche einen der PGP 5/6 kompatiblen WWW-Keyserver auf, wähle die Seite aus, wo Keys an den Server versendet werden können, und füge in das Formularfenster die Datei "MeinDSS/DH.asc" ein.

Versand eines Keys aus PGPkeys heraus

bei PGP 5

1. PGPkeys aufrufen
2. Public Key markieren
3. im Menü "Keys" den Eintrag "Keyserver" auswählen
4. "Send Selected Keys"
5. oder markiere den Public Key
6. mit 1xMKrT Kontextmenü aufrufen
7. Eintrag "Keyserver"
8. "Send Selected Keys" wählen

bei PGP 6

1. PGPkeys aufrufen
2. Public Key markieren
3. im Menü "Server" den Eintrag "Send to" auswählen
4. gewünschten Keyserver anklicken
5. oder markiere den Public Key
6. mit 1xMKrT Kontextmenü aufrufen
7. Eintrag "Send to"
8. gewünschten Keyserver anklicken

Homepage

Der exportierte Public Key liegt als "MeinRSA(DSS/DH)PublicKey.asc" im Homepageverzeichnis. Mit einem Link in der Form "http://Verzeichnis/Dateiname.asc" kann sich jeder, der die Homepage besucht, den Schlüssel direkt herunterladen

Newsgroup

Der exportierte Public Key ("Dateiname.asc") kann in ein Posting gesetzt und in die newsgroup "z-netz.alt.pgp.schluessel" gesendet werden.
Entweder lässt man sich "Dateiname.asc" in einem Dateibetrachter oder Editor anzeigen, markiert den kompletten Inhalt zwischen -----BEGIN PGP PUBLIC KEY BLOCK--- und ---END PGP PUBLIC KEY BLOCK--- und fügst in dann in den Body des Postings ein oder man importierst die Datei direkt über das Mail/Newsprogramm in den Body des Postings.

E-Mail Signatur

In der Signatur, die man in Postings oder E-Mails verwendet kann ein Hinweis auf die Existenz des Public Keys eingefügt werden.

Hinweis

Die Netiquette empfielt eine Länge von 4 Zeilen mit XY Zeichen vor, die für eine Signatur verwendet werden soll.

Beispiele

- (RSA/DH)PGP-Key unter http://Verzeichnis/Dateiname.asc
- (RSA/DH)PGP-Key auf Anfrage, PGP-Key on request
- (RSA/DH)PGP-Key Server ID: 0xXXXXXXXX

Anmerkung

Es ist sinnvoll immer kenntlich zu machen, um welchen Keytyp es sich handelt: RSA-Key oder Diffie-Hellmann-Key. Es ist nicht sinnvoll, in einem eher öffentlichen Raum wie einer Newsgroup oder einer Mailingliste den Public Key als Fileattachement anzuhängen, da so auch alle, die nicht an dem Public Key interessiert sind, ungewollt in den "Genuss" des Public Keys kommen würden. Man braucht auch nicht den gesamten öffentlichen Schlüsselbund (pubring.pkr) zu versenden ! Darüber wird sich der Empfänger freuen, wenn das nicht passiert. Ausserdem braucht niemand zu wissen, mit wem man per E-Mail und PGP korrespondiert.

Achtung: Ein einmal auf einem Keyserver hinterlegter Public Key kann nicht mehr als ungültig/zurückgezogen erklärt werden, wenn keine Rückzugsurkunde (Revocation) für diesen Key erstellt wurde, bzw. verfügbar ist oder die Passphrase in Vergessenheit geraten ist !

Schlüsselanforderung oder
"Wie gelange ich an die Public Keys anderer Personen ?

In dem vorherigen Kapitel wurden ja schon alle Quellen genannt, über die ebenso die Public Keys anderer Personen zu beziehen sind: per E-Mailanforderung bei der Person, über dessen Homepage, in der Newsgroup oder über die Keyserver.
An dieser Stelle soll das Holen eines Public Keys per Keyserver im Vordergrund stehen.
Als Voraussetzung zum Einholen eines Keys muss mindestens einer der unten aufgeführten Bestandteile des Keys bekannt sein, der benötigt wird: Das kann die User-ID (oder ein Bestandteil) des Keybesitzers, die Key-ID des Schlüssels oder die E-Mailadresse sein.

per E-Mail

Zu einem Keyserver (z. B. pgp-public-keys@informatik.uni-hamburg.de) kann eine E-Mail mit den folgenden Subjects gesendet werden, um einen Key anzufordern, ohne noch etwas zusätlich in den Body zu schreiben:

GET (User-ID) oder (E-Mail Adresse) oder (Key-ID mit Syntax "OxXXXXXXXX"), wenn die genauen Daten vorliegen

oder

MGET (einem Bestandteil von User-ID/E-Mail Adresse), wenn nur ungefähre Angaben bekannt sind

über ein WWW-Interface

Wenn zum Beispiel der WWW-Keyserver der Universität Paderborn angewählt wird, kann über ein bequemes WWW-Interface nach Public Keys gesucht und die gefundenen Keys angezeigt werden lassen ("Extract A Key"). Über das gleiche Interface können auch Keys an den Keyserver versenden werden ("Submit A Key"). Formulareingabefelder zur Eingabe eines Suchstrings und Optionen zur Ausführlichkeit der Anzeige stehen dort bereit.

aus PGPkeys heraus

bei PGP 5

1. Aufruf von PGPkeys
2. Menü "Keys"
3. Eintrag "Keyserver - Find a new key"
4. E-Mail Adresse oder User-ID des gewünschten Keys eintragen

bei PGP 6

1. Aufruf von PGPkeys
2. Menü "Server"
3. Eintrag "Search"
4. im Dropdownmenü gewünschten Keyserver auswählen
5. Suchkriterien über "More Choices" festlegen und erweitern
6. bekannte Daten eingeben

Danach nimmt PGP zu dem Keyserver, der in den Optionen eingetragen ist, Verbindung auf und ladet den Key herunter, nachdem man die Nachfrage seitens PGP, ob man den Key hinzufügen mit Ja bestätigt hat.

Über die PGP 5/6 WWW-Keyserver

Die PGP 5/6 WWW-Interfaces bedient man genauso wie die WWW-Interfaces der RSA-Keyserver.

Möglichkeiten

a) wenn ein Public Key über das Keyservermenü von PGP 5/6 angefordert wurde, wird er von PGP selbst in den Pubring aufgenommen.
b) bei Zusendung per E-Mail durch den Besitzer oder durch einen Keyserver kann der Body des Textes, der den Key enthält, kopiert werden, anschliessend klickt man auf PGPtray und wählt die Option "Add Key from Clipboard". Genauso kann man verfahren, wenn der Key Bestandteil einer Webpage ist.

Wird der Key als Fileattachement mitgeschickt, liegt er als Datei "PublicKey.asc" in einem Verzeichnis.
In diesem Fall PGPkey aufrufen und den Key über das "Keys"-Menü importieren.
Ebenso wird verfahren, wenn man sich den Public Key von einer Homepage des WWW holt, sobald der Key als Datei auf der heimischen Festplatte abgespeichert ist.

Aufnahme eines neuen Public Keys bei **PGP 2.6.3**

```
PGP 2.6.3> pgp -ka Datei-mit-Public-Key
```

Weitere PGP Keymanagementfunktionen oder
"Was gibt es sonst noch ?"

Das "An- und Abschalten" (Disable/Enable) eines Keys

Mit PGP können einzelne Public Keys an- und abgeschaltet werden, wenn man nicht ständig alle fünfzig Public Keys benötigt, die sich im Pubring befinden. Dazu wird der betreffende Key im PGPkeys-Window markiert und dann über den Menüpunkt "Keys - Disable" oder durch Wahl des gleichen Menüpunktes im Kontextmenüs (Aufruf durch 1xMKrT) abgeschaltet.
Im Window werden die abgeschalteten Keys grau dargestellt, User-ID und E-Mailadresse in kursiver Schrift. Der Effekt zeigt sich beim Signieren und Verschlüsseln, denn dann werden nur die angeschalteten Keys aufgelistet, was Übersicht und Auswahl erleichtert. Genauso werden umgekehrt die abgeschalteten Keys mit dem Menüpunkt "Enable" wieder angeschaltet.

An- und Abschalten von Public Keys bei **PGP 2.6.3**:

```
PGP 2.6.3> pgp -kd User-ID
```

Der "Default"-Key

Verwendest Du mehrere Public Keys oder einen RSA- und einen DSS/DH-Public Key kannst Du einen Key zu Deinem Standardschlüssel erklären. Der Standardschlüssel ist im PGPkeys-Window daran zu erkennen, dass User-ID und E-Mailadresse in fetter Schrift erscheinen.
Die Vorteile:

1. wenn Du ausgehende E-Mails immer auch an Dich selbst verschlüsselst, wird Dein Standardkey automatisch in die Empfängerliste gesetzt.
2. Bei der Signierung erscheint der Standardkey im Keyauswahlmenü an erster Stelle.

Man setzt einen Public Key als Standard, indem der ausgewählte Key markiert und dann über das "Keys"-Menü oder das Kontextmenü der Eintrag "Set As Default" angeklickt wird.

Der Default Public Key bei **PGP 2.6.3**:
wird in der **config.txt** über den Eintrag **MyName="Eigene User-ID"** gesetzt

Abgetrennte Signaturen

Normalerweise setzt man eine PGP Signatur unter einen E-Mailtext. Es kann aber erforderlich sein, eine PGP Signatur zu einer einzeln vorliegenden Datei zu erzeugen, ohne das diese Datei selbst verändert wird (z. B. möchte man eine HTML-Datei versenden, aber die Seite soll im Browser ohne PGP-Signatur erscheinen oder es handelt sich um Binärdateien). Das heisst, die Erzeugung einer abgetrennten Signatur kann nur anhand einer schon vorliegenden Text- oder Binärdatei durchgeführt werden.
Über die Zwischenablage oder die E-Mail Plug-Ins ist dies nicht möglich.

Vorgehensweise

1. die Datei wird im Dateimanagerfenster (z. B. Explorer) markiert und über den PGP-Eintrag des Kontextmenüs wird die Signierfunktion ("Sign") aufgerufen

2. Eingabe der Passphrase
3. Unter "Options" wird die Option "Detached signature file" aktiviert.
   Soll die Signatur als Textform in ASCII-Zeichen erstellt werden, auch die Option "Text Output"
4. Abspeichern der Signatur als eigene Datei "Signaturdatei.sig" (dabei wird als Dateiname immer der Dateiname + Endung der zu signierenden Datei vorgeschlagen)

Abgetrennte Signatur bei **PGP 2.6.3**:

```
PGP 2.6.3> pgp -sb Datei Eigene-User-ID
```

Der Empfänger kann jetzt anhand der Signaturdatei überprüfen, ob die Originaldatei unverfälscht und unverändert zu ihm gelangt ist und wer der Absender der Datei ist.

Vorgehensweise

1. die Datei "Signaturdatei.sig" markieren
2. über den PGP-Eintrag des Kontextmenüs die angezeigte Funktion "Verify Siganture" aktivieren
3. in dem nun aufklappenden Fenster die Originaldatei markieren

PGP blendet jetzt eine Informationsmeldung ein, die abhängig davon ist,
ob Originaldatei und Signaturdatei zusammenpassen oder nicht:

> "Good signature from (User-ID E-Mailadresse). Signature made on (Tag und Datum der Signaturerstellung)"
>
> oder
>
> "Bad signature from (User-ID E-Mailadresse). Signature made on (Tag und Datum der Signaturerstellung)"

Die Änderung der Passphrase

Der Wert und die Bedeutung der Passphrase sollte an dieser Stelle bekannt sein. Andernfalls verweise ich auf die Informationen der Mantra FAQ.
So ist leicht einzusehen, dass die Passphrase einer der Angriffspunkte darstellt, an denen Hacker, Geheimdienste oder Spinner ansetzen werden, um den eigenen kryptographischen Schutz zu durchbrechen. Aus diesem Grund kann es sinnvoll sein, die Passphrase in Abständen zu ändern.

Vorgehensweise

1. PGPkeys aufrufen
2. das »Key Properties« Fenster aktivieren
3. den »Change Passphrase« Button anklicken
4. alte Passphrase eingeben
5. neue Passphrase eingeben
6. neue Passphrase zur Bestätigung wiederholen
7. OK

Änderung der Passphrase bei **PGP 2.6.3**:

```
PGP 2.6.3> pgp -ke Eigene-User-ID
```

Die Änderung der User-IDs des Public Keys

Auch mit PGP 5./6.X ist möglich, die User-ID, also "Name <E-Mailadresse>", eines Keys zu ändern.
Oder es kann eine weitere, neue User-ID hinzugefügt und eine bestehende User-ID gelöscht werden.

A. Hinzufügen einer User-ID

Vorgehensweise

1. PGPkeys aufrufen
2. Key markieren
3. 1xMKrT oder Aufruf des Menüs "Keys"
4. Menüpunkt "Add Name" wählen
5. Eingabe von "Name <E-Mail-Adresse>"
6. Eingabe der Passphrase

bei **PGP 2.6.3**:

```
PGP 2.6.3>pgp -ke Eigene-User-ID
```

B. Löschen einer User-ID

Vorgehensweise

1. PGPkeys aufrufen
2. Key über 1xMKIT auf das + Zeichen vor dem Key
3. Betreffende User-ID markieren
4. 1xMKrT oder Aufruf des Menüs "Keys"
5. Menüpunkt "Delete" wählen

bei **PGP 2.6.3**:

```
PGP 2.6.3> pgp -kr User-ID LW:\keyring.pgp
```

C. Änderung der eigenen ersten User-ID

Vorgehensweise

1. Neue User-ID erzeugen wie unter A. beschrieben
2. 1xMKrT oder Aufruf des Menüs "Keys"
3. Menüpunkt "Set As Primary Name (User-ID)" wählen

Anmerkung

Erschien im PGPkeys Fenster zuerst »test <test@test>« als primäre User-ID und lautete die neue User-ID »test2 »test2@test2.de>«, erscheint nach dem Vorgang die neue User-ID als die primäre User-ID, die ursprüngliche User-ID wird darunter angefügt, wie alle weiteren User-IDs und die ursprünglichen Signaturen der nachfolgenden User-IDs (die auf test <test@test> lauteten) werden alle durch die Signatur der neuen User-ID ersetzt, d. h. alle nachfolgenden User-IDs sind mit test2 <test2@test2.de> signiert.

Hinzufügen einer "Foto User-ID" (nur bei PGP 6 DH/DSS Keys)

Um dem Key sein eigenes Foto hinzuzufügen nimmt man ein Passfoto, verkleinert es nach dem Einscannen auf 120 x 144 Pixel und speichert es als BMP oder JPG Datei.

Vorgehensweise:

1. PGPkeys aufrufen
2. DH/DSS Key markieren
3. über das Menü "Keys" oder durch 1 MKrT im Kontextmenü "Add...Photo" aufrufen
4. die Bilddatei aus dem Explorer in den Rahmenbereich ziehen oder über den Button "Select File" die Bilddatei öffnen
5. Passphrase eingeben

Die gleiche Vorgehensweise wird zum Löschen und Ändern einer Foto User-ID gewählt

Hinzufügen von "Designated Revokers" (nur bei PGP 6 DH/DSS Keys)

Ein Designated Revoker ist eine zweite Person, die vom Keybenutzer dazu ermächtigt (designated) ist, für den den eigenen Public Key eine Rückzugsurkunde (Revocation) auszustellen und ihn als zurückgezogen (revoked) auf dem Keyserver zu kennzeichnen.

Achtung:
Wenn einmal einem Public Key ein Designated Revoker hinzugefügt wurde, ist der Prozess nicht mehr umkehrbar, der Designated Revoker kann nicht wieder vom Key entfernt werden.
Natürlich setzt die Vergabe des Rechts an eine andere Person, den eigenen Public Key als ungültig zu erklären, 100% Vertrauen in diese Person voraus.
Eine m. M. nach bessere und sicherere Methode, den eigenen Key bei Verlust von Secret Key und/oder Passphrase dennoch revoken zu können, findet sich im Kapitel Key Revocation

Vorgehensweise:

1. PGPkeys aufrufen
2. den eigenen DH/DSS Key markieren
3. über Menü "Keys" oder 1MKrT im Kontextmenü "Add...Revoker" auswählen
4. die Key User-ID der zu ermächtigenden Person auswählen
5. den Bestätigungsdialog bejahen
6. die Passphrase eingeben
7. Kopie des Public Keys an den Revoker und an den Keyserver senden

Wiederholungsüberprüfung der Signaturen (Reverify Signatures) (nur PGP 6)

Über das Menü Keys Reverify Signatures werden alle Signaturen eines Public Keys auf Gültigkeit (Validity)hin überprüft.

Überprüfung der Signaturen bei **PGP 2.6.3**:

```
PGP 2.6.3> pgp -kc oder -km User-ID
```

Das Zerlegen des Secret Keys auf mehrere Keyteilinhaber (Share Split/Key Split) (nur PGP 6)

Seit PGP 6 kann der RSA oder DH Secret Key eines Key Paares in mehrere Teile (Shares) zerlegt (Splitting) und die Teile mit dem Public Key anderer Personen oder mittels Vergabe von Passphrases konventionell verschlüsselt als Datei "User-ID Share.shf" abgelegt werden (Blakely-Shamir Key Splitting). Damit die Keyteile erstellt werden können, müssen alle Keyteilinhaber anwesend sein, um ihre Passphrase festlegen oder eingeben zu können, bei Public Key Verschlüsselung müssen die Public Keys der Keyteilinhaber im Pubring vorhanden sein. Zum Entschlüsseln oder Signieren einer Datei/Nachricht werden die Keyteile nach Eingabe der Passphrases wieder temporär zusammengesetzt.
Key Splitting ist in Bereichen höherer Sicherheitsanforderungen sinnvoll:
Zum Beispiel gibt es die bekannte Taktik von Guerillagruppen, sich in mehrere Zellen aufzuteilen, damit bei einer Verfolgung oder Verhaftung nicht die gesammte Gruppe und ihre Materialien in die Hände der Gegenpartei fallen. Würde PGP in einer Guerillagruppe eingesetzt werden, so würde jeder Zellenführer einen Keyteil erhalten. Wenn jetzt ein Zellenführer verhaftet und sein Keyteil in die Hände der Verfolger fallen würde, könnte die gesammte Gruppe weiterhin Dokumente signieren und entschlüsseln, die Verfolger aber nicht.
Das Prinzip kann man sich auch ähnlich vorstellen wie die mehreren Tresorschlüssel, die Bankangestellte gemeinsam benutzen müssen, um die Tresorkombination herzustellen, damit sie den Tresor öffnen oder verschliessen können.

Die Herstellung eines Key Splittings:

1. zuerst wird ein vorhandener Key ausgewählt oder ein neuer Key hergestellt.
2. der aufzuteilende Key wird markiert
3. Aufruf von "Share Split" über das Menü "Keys" oder über das Kontextmenü mit 1MKrT
4. es öffnet sich das "Split Key" Konfigurationsfenster:
    - im Feld "Split Key" ist die User-ID des ausgewählten Keys zu sehen
    - unter "Shareholders" (Keyteilinhaber) werden die User-IDs der Keyteilinhaber angezeigt, wenn die Keyteile mit vorhandenen Public Keys der Keyteilinhaber verschlüsselt werden oder die Namen der Keyanteilinhaber bei rein konventioneller Verschlüsselung der Keyteile.
    - unter "Total Shares Required to Decrypt or Sign" (Anzahl der Keyteile, die zum Entschlüsseln oder Signieren nötig sind) wird die gewünschte Anzahl der Keyteile (bis zu 99) eingegeben.
      Es können auch mehr Keyteilinhaber als Keyteile selbst angegeben werden. Wenn man z. B. den Key auf drei Keyteilinhaber aufteilt, die erforderliche Anzahl der Keyteile zum Entschlüsseln/Signieren jedoch auf zwei beschränkt, bietet das den Vorteil, dass eine Datei/Nachricht immer noch entschlüsselt oder signiert werden kann, obwohl ein Keyteilinhaber seine Passphrase vergessen hat oder eine Keyteildatei abhanden gekommen ist.

- Aufteilen an Public Key Besitzer
    1. die entsprechenden User-IDs der Keys per Drag and Drop aus dem PGPkeys Fenster in das Shareholders Fenster ziehen
    2. auf den Button "Split Key" klicken
    3. Verzeichnis auswählen, in dem die Keyanteildateien abgelegt werden sollen
    4. ursprüngliche Passphrase des zu teilenden Keys eingeben
    5. Bestätigungsmeldung bejahen
    6. anschliessend befinden sich die Keyanteildateien im angegebenen Verzeichnis

- Aufteilen an Personen ohne Public Key (konventionelle Verschlüsselung der Keyteildateien)
    1. über den Button "Add" den Namen des Keyteilinhabers eingegeben
    2. anschliessend muss jeder Keyteilinhaber eine Passphrase für seinen Keyanteil eingeben
    3. sind alle Keyteilinhaber aufgelistet, den Button "Split Key" anklicken
    4. Verzeichnis für die Keyanteildateien auswählen
    5. ursprüngliche Passphrase des Keys eingeben
    6. Bestätigungsdialog bejahen

Das Verschlüsseln, Entschlüsseln und Signieren mit Split Keys

Verschlüsseln:

eine Datei oder Nachricht wird mit dem Split Key genauso verschlüsselt wie mit einem herkömmlichen Public Key

Entschlüsseln & Signieren:

1. Aufruf der Entschlüsselungs- oder Signierfunktion von PGP nach bekannter Weise
2. im Fenster "Key Share Collection" über den Button "Select Share File" die Keyteildateien öffnen
3. jeder Keyteilinhaber gibt seine Passphrase ein
4.
    - bei Entschlüsselung kann das Chiffrat entschlüsselt als Datei abgespeichert, bzw. als Clipboardinhalt eingefügt werden
    - bei Signierung kann das Chiffrat als signierte Datei abgespeichert, bzw. als Clipboardinhalt eingefügt werden

Das Wiederherstellung des geteilten Keys

1. über das Menü "Keys", "Properties" den Button "Join Key" anklicken
2. für jeden Keyteil über "Select Share File" die Keyteildatei öffnen
3. Keyteilinhaber muss seine Passphrase eingeben
4. anschliessend die ursprüngliche oder neue Passphrase für den wieder zusammengefügten Key eingeben

Web of Trust oder
"Wie funktioniert das Netz des Vetrauens ?"

Wenn jemand Public Keys dazu benutzen will, um verschlüsselte Kommunikation zu betreiben, muss er zwei Dinge voraussetzen können:

1. die Public Keys seiner Kommunikationspartner sind echt.
2. sein Public Key kann nicht gefälscht werden.

Deshalb fängt das Web of Trust schon bei der Schlüsselerzeugung an.

Fälschung

Bei der Keyerzeugung wird der eigene Public Key mit dem eigenen Secret Key signiert. Wenn man unter PGPkeys nachsieht, befinden sich unter dem Keysymbol zwei weitere Einträge, dem ersten Eintrag ist ein symbolisierter Briefumschlag vorangestellt und steht für die User-ID, dem zweiten Eintrag ist eine Feder oder eine Stiftspitze vorangestellt ist und steht für die Signatur. Sieht man sich mal die "Key Properties (Eigenschaften)" des eigenen Public Keys an, fällt eines auf:
Sowohl die "Key-ID" als auch der "Key-Fingerprint" sind bei dem Keypaar, der User-ID und der Signatur gleich.
Kontrollieren wir den Public Key eines Kommunikationspartners (was immer der Fall sein muss), **muss** das Ergebnis genauso aussehen, ansonsten ist der Public Key gefälscht. PGP unterschreibt also selbst unseren eigenen Public Key, um damit auszuweisen, dass der Public Key mit dieser speziellen Key-ID und User-ID mit dem richtigen Secret Key unterschrieben wurde, der Verbreiter des Public Key mit User-ID wx und Key-ID yz im Besitz des passenden Secret Keys und der passenden Passphrase ist. Ausserdem beinhaltet die eigene Signatur weitere Merkmale des Schlüssels wie z. B. die Gültigkeitsdauer, die im Fingerprint des Schlüssels nicht enthalten sind und bestätigt die Zugehörigkeit des Keyinhabernamens in der User-ID wx zum Schlüssel mit der Key-ID "yz".
Die ID's der Signatur sind nicht zu editieren, wie es beim Public Key möglich ist, so dass der Key durch die Eigensignatur auch vor Veränderungen (mit Ausnahme weiterer Signaturen anderer User, die ich nicht verhindern oder nochmals signieren kann) geschützt wird.

Fälschungswege

A.
Der Fälscher versieht den ursprünglichen Public Key mit seiner User-ID/E-Mail Adresse und um nicht aufzufallen, entfernt er alle Signaturen.
Da die Keyserver nur einmal einen Key mit einer spezifischen Key-ID speichern können, kann der Fälscher versuchen, den Key vor dem ursprünglichen Key des Erstellers auf dem Keyserver abzulegen, so dass der Ersteller nicht mehr in der Lage ist den ursprünglichen, erzeugten Key zu hinterlegen. Die Fälschung fällt auf, da der Key keine Signatur hat. Denkbar ist, dass der Fälscher eine Signatur erstellt und deren Fingerprint fälscht, dann aber unterscheidet sich der gefälschte Key vom originalen, signierten Key in der Länge.

B.
Der Fälscher erstellt einen neuen Public Key, der die User-ID und die Key-ID des ursprünglichen Keys des Angegriffenen trägt und versieht den Key mit seiner Signatur und zusätzlich eventuell noch mit seiner E-Mail Adresse. Aus der Sicht des Fälschers hofft dieser, dass Personen annehmen werden, dieser Key gehoere dem originalen User und diesen Key benutzen um E-Mails an den eigentlichen Empfänger verschlüsseln. Der Fälscher kann sich nun zwischen dem gutgläubigen Absender und den eigentlichen Empfänger schalten, die Korrespondenz abfangen und entschlüsseln.

Ist die E-Mail Adresse des Keys durch die Adresse des Fälschers ersetzt worden, kann er sich diesen Schritt sogar sparen und bekommt die E-Mails direkt zugeschickt. Andernfalls erhält der ursprüngliche Empfänger zwar die E-Mails, kann diese aber nicht mehr entschlüsseln. Wurde der Fingerprint nicht gefälscht, ist die Fälschung am unterschiedlichen Fingerprint von Key und Signatur zu erkennen.
Weitere Informationen, welche Gefahren durch Fälschungsmöglichkeiten entstehen können, finden sich in der:

- PGP Attacks FAQ
- Comp.Security.PGP FAQ
- Eigensignatur (Selfsign) FAQ

Was deutlich wird, ist, dass man nur dann ziemlich sicher von der Echtheit eines Keys ausgehen kann, wenn die **Kombination aus Keylänge, die ID's, die Fingerprints und die Signatur** stimmt und diese Kombination persönlich beim Keybesitzer überprüft wurde.

Somit kann zwar nicht ausgeschlossen werden, dass jemand einen Public Key fälscht, aber, dass er dies unentdeckt tun kann.

Validity

Die Echtheit ("Validity")eines Public Keys wird durch die Signatur anderer User "bezeugt". Wir können einen anderen Public Key mit unserer eigenen Signatur versehen. In PGP 5/6 wird dieser Key sogleich "valid", d.h. "gültig" oder "authentisch". Solange er nicht von uns signiert ist, bleibt der Key grundsätzlich gesehen "ungültig" oder "nicht authentisch". Als Folge bleibt der Besitzer des Keys auch als "untrusted" oder "nicht vertrauenswürdig" eingestuft. Mit der Signierung bezeugen wir, dass wir uns überzeugt haben, dass dieser Public Key wirklich zum angegebenen User gehört oder anders: Wir wissen mit 100% Sicherheit, dass der Public Key echt ist, unabhängig davon, ob unter diesem Key bereits Signaturen von Leuten stehen, denen wir vertrauen oder die uns bekannt sind. Zu diesem Zweck überprüfen wir schon mal, wie oben angegeben, den bereits vorliegenden Public Key auf Unstimmigkeiten.
Dann überprüfen wir persönlich oder telefonisch die Angaben, die uns der Key liefert (User-ID, Key-ID, Fingerprint, Keylänge, Keytyp) anhand der Angaben, die uns der Kontaktierte auf Anfrage liefert. Zusätzlich können auf die gleiche Weise eine oder mehrere der unter dem Key befindlichen Signaturen gegengeprüft werden.
Wenn im Idealfall, davon ausgegangen wird, dass **jeder PGP-Benutzer** so verfährt, wären idealerweise alle Public Keys echt. Da wir die Realität kennen, kann sich **jeder PGP-Benutzer** ausrechnen, welche Kontrollen er durchführen muss und wie verantwortlich er zu handeln hat.
Neben den Signaturen anderer User spielen die Zertifikate von CA's eine weit wichtigere Rolle. Gesetzlich anerkannte CA's müssen dem Signaturgesetz und der Signaturverordnung entsprechen, d. h. auf gesetzlich vorgeschriebenem Wege arbeiten und Zertifikate ausstellen.
Dehalb besitzen CA-Zertifikate einen verbindlicheren und vertrauenswürdigeren Charakter als Signaturen anderer User und man sollte bemüht sein, wenigstens ein Zertifikat einer anerkannten CA für den eigenen Key zu erhalten.

Trust

Neben der "Validity" hat PGP 5/6 auch eine weitere interne "Trust"-Skala zu bieten, die sich auf unsere Einschätzung des Keybesitzers bezieht. Der "Trust" oder "Vertrauensgrad" gibt in drei Stufen von "untrusted-nicht vertrauenswürdig" über "marginal-begrenzt vertrauenswürdig" bis zu "complete-voll vertrauenswürdig" an, inwieweit das Vertrauen in das verantwortliche Handeln und die Fähigkeit eines Keybesitzer vorhanden ist, selbst wiederum für die "Validity" eines anderen Public Keys zeugen zu können.

Bekommt man irgendwann einen Public Key, der mit der Signatur einer Person versehen ist, die man selbst eines von Dir nach "Trust" eingestuften PGP-Benutzers signiert ist, wird dieser Public Key entsprechend als gültig eingestuft, auch wenn Du diesen neuen Key selbst noch nicht signiert hast.
Beachte: Der Trustlevel eines PGP-Benutzers kann erst höher als "untrusted" eingestuft werden, wenn Du den Key dieses Benutzers selbst signiert hast. So hängt also das Vetrauen in einen User von der Gültigkeit seines Keys ab. Oder anders formuliert: Ein Key kann zwar "gültig", der Benutzer aber "nicht vertrauenswürdig" sein.

Die **Konfiguration des "Web of Trust" bei PGP 2.6.3**: wird durch die Vergabe der eigenen Signatur und/oder der Vergabe eines von vier Trustlevels bestimmt.
PGP 2.6.3 kennt zur eigenen Kennzeichnung des Vertrauens in die Aufrichtigkeit der Person und in die Kompetenz einer Person, verantwortlich sowohl mit den eigenen, wie mit anderen PGP-Keys umzugehen, folgende Stufen:

- 1 = ich weiss nicht
- 2 = Nein
- 3 = in der Regel
- 4 = Ja, immer

über die man der Datei **config.txt** in zwei Einträge den eigenen Bereich des Web of Trust regeln kann:

1. **Completes_Needed** = X
   PGP sieht dann einen Public Key als echt an, wenn er X Signaturen trägt, deren User die Vertrauensstufe 4 besitzen
2. **Marginals_Needed** = Y
   PGP sieht dann einen Public Key als echt an, wenn er Y Signaturen trägt, deren User die Vertrauensstufe 3 besitzen

Zur direkten Einstellung des Trustparameters bei PGP 2.6.3 für einen User:

```
PGP 2.6.3> pgp -ke User-ID
```

Zertifizierungsstellen ("Certification Authorities")

Einen weiteren Ansatz stellen die sogenannten Zertifizierungsstellen ("Certification Authorities"/CA) dar, also anerkannte Institutionen, Organisationen, Firmen oder Vereine, die die Zugehörigkeit zwischen der Identität einer Person und einem Public Key nach bestimmten Prüf- und Kontrollverfahren durch ein Zertifikat zertifizieren. Damit entfällt der doch etwas aufwendige und mit Unsicherheiten belastete Weg der Sammlung von Signaturen unter den eigenen Public Key und die Notwendigkeit der persönlichen Kontaktaufnahme mit dem Keybesitzer, wenn man einen anderen Public Key zertifizieren will.
Aber auch bei den CA's ist es meistens notwendig, persönlich mit dem Personalausweis oder gar einer notariellen Bestätigung zu erscheinen, damit diese den eigenen Public Key zertifizieren oder das Keypaar muß in einem abgesicherten Verfahren direkt in der CA vom Keybesitzer erzeugt werden.
Es gibt aber auch Zertifikate, die dann ausgestellt werden, wenn der Keyinhaber einen verschlüsselten Prüfsatz, den ihm die CA nach Antrag zusendet, entschlüsseln kann und dann wieder per E-Mail an die CA zurücksendet (E-Mail Zertifikat) oder die CA verlangt zusätzlich die Ausweisung gegenüber einer bekannten öffentlichen Stelle wie einem Postamt (Post-Ident-Zertifikat).

Es gibt also unterschiedliche "Qualitätsstufen" bei Zertifikaten, die aber auch aus der Bezeichnung eines Zertifikats ersichtlich sind. Der Vorteil der CA-Signatur liegt trotzdem im allgemeinen Bekanntsheits- und Vertauensgrad, der es leichter macht, einem Public Key zu vertrauen, der mit einer CA-Signatur versehen ist, als einem Public Key, der zwar von einer Person signiert wurde, deren Name mir aber nicht bekannt ist.
Die Bedeutung der CA's wird deshalb mit einer steigenden Anzahl von PGP Benutzern zunehmen.

Bekannte CA's

- die PGP-CA der Zeitschrift c't
- die PGP 2.6.3in-CA des Individual Networks
- die DFN-CA des Deutschen Forschungsnetzes
- die Firma TC TrustCenter for Security
  TrustCenter bietet eine für Privatpersonen kostenlose Zertifizierung per E-Mail ("EMail Certificate") oder in Kombination mit einer Überprüfung per Einschreiben ("Certified Mail-Certificate") an.
  Zum Ende des Jahres können auch Keys mit Pseudonym User-ID's zertifiziert werden.
- die CA des GeFoekoM e.V. / AK Datenschutz

Weitere Informationen zur PGP-Zertifizierung und Kurzinfos zu den angegebenen CA's finden sich über **Skylla: PGP-Schlüssel-Zertifizierung** von Alexander Svensson

Auch der **Staat** denkt an die Einrichtung dieser Zertifizierungsstellen, in eigener Regie und durch Lizenzvergabe, allerdings ist diese Idee oft mit der Hinterlegung eines "Generalschlüssels" bei staatlichen Stellen oder **"Trusted Third Parties" (TTP) oder "TrustCentern"**, dem sogenannten **Key Escrow** (im Falle von PGP könnte man auch sagen: Hinterlegung von Secret Key und Passphrase) verbunden, mit dessen Hilfe staatliche Organe, wie die Geheimdienste, nach einem juristischen Genehmigungsverfahren verschlüsselte Dateien oder E-Mails bei Verdacht wieder entschlüsseln können oder mit der Zulassung von Verschlüsselungsprogrammen, die es erlauben, den Verschlüsselungskey, der bei der Verschlüsselung benutzt wurde, wiederherstellen zu können, sogenanntes **Key Recovery**.
Eine Zwischenform stellen CA's dar, die gleichzeitig die Merkmale der TTP's besitzen, d. h. das Schlüsselpaar wird in der Zertifizierungsstelle erzeugt und dem Antragsteller auf einer SmartCard ausgehändigt, der Public Key wird in ein öffentliches Verzeichnis eingestellt, aber der private Schlüssel verbleibt (wenigstens für den Zeitraum der Schlüsselerzeugung) ebenfalls in der Zertifizierungsstelle und macht sie so zur TTP - dazu zählen z. B. die D-Trust GmbH und die Telesec.
TTP´s und ihre Mischformen sind grundsätzlich abzulehnen, da der Private Key in der Einrichtung verbleibt. Bei dieser Form aus CA und TTP/TrustCenter kommt mir persönlich ein ganz anderer Verdacht...und Carl Ellison von der Firma CyberCash nennt diese Vorstellungen beim Namen: **"Govermental Access to Keys (GAK)"**.
Daneben muss für die Sicherheit des Public Keys, Secret Keys und der Passphrase gesorgt werden. Eine sichere Verwahrung von Pubring, Secring und Passphrase gehören dazu. Ein zusätzlicher Sicherheitsgewinn kann die wöchentliche oder monatliche Änderung der Passphrase sein, was ein Errechnen/Erraten der Passphrase zusätzlich erschwert, wenn man davon ausgeht, dass dieser Vorgang einige Zeit und einige Rechnerkapazitäten voraussetzt.
Es ist auch sinnvoll, sofort nach der Erstellung von Public und Secret Key, die Schlüsselrückzugsurkunde ("Key Revocation") für den Public Key zu erzeugen und gesichert abzulegen. Das hat den Vorteil, dass sofort die Key Revocation an einen Keyserver gesendet und die Kommunikationspartner informiert werden können, wenn Secret Key und/oder die Passphrase in die Hände anderer gelangt ist (was immer geschehen muss, wenn dieser Fall eintreten sollte).

Was noch wichtiger ist: Wenn der Keybesitzer aus irgendeinem Grund nicht mehr im Besitz des Secret Keys und der Passphrase ist, kann er seinen Public Key zurückziehen.

Schlüsselzertifikate oder
"Wie signiere ich die Public Keys anderer Personen ?"

Voraussetzung für das Signieren anderer Public Keys ist das Lesen des Kapitels "Web of Trust".

Vorgehensweise

1. Aufruf von PGPkeys
2. a) Menü "Keys - Sign" oder
   b) 1xMKrT auf den betreffenden Key und im Kontextmenü "Sign" wählen
3. Button "More Choices" anklicken

Signiermöglichkeiten (Signature Type)

Non-Exportable

bei der nicht exportierbaren Signatur verbleibt die Signatur nur im eigenen Pubring, sie wird nicht mit übertragen, wenn man den Key abspeichert, in eine E-Mail Nachricht einfügt oder an einen Keyserver sendet. Diese Signatur sollte man wählen, wenn man z.B. schon einen dauerhaften E-Mailverkehr mit einer Person pflegt und sich daraufhin *ziemlich sicher* ist, dass der Key dem Keybesitzer gehört und deshalb im PGPkeys Window und den Bestätigungsfenstern den Key als echt ausgewiesen sehen möchte.

Exportable

die exportierbare Signatur ist die eigentliche Signatur im herkömmlichen Sinne, wie wir sie auch von PGP 2.6.3 kennen.
Sie wird beim Abspeichern eines Keys, beim Versenden des Keys per E-Mail oder an einen Keyserver mit übertragen und bestätigt somit gegenüber der Öffentlichkeit, dass der Key wirklich dem Keybesitzer gehört und nicht gefälscht ist.

Aus diesem Grund sind bei Wahl der exportierbaren Signatur die Prinzipien des "Web of Trust" unbedingt zu beachten und anzuwenden !

Meta-Introducer Non-Exportable und Trusted Introducer Exportable

durch unsere Signatur wird der Key eines Meta-Introducers (oberster, vertrauenswürdiger Bürge) und die Keys, die der Meta-Introducer signiert zu Trusted-Introducers (vertrauenswürdiger Bürge). D. h. der Meta-Introducer erhält unser volles Vertrauen (Trust) und sein Key volle Gültigkeit (Validity).
Die Signatur des Meta-Introducers ist nicht exportierbar, da er eine herrausragende Stellung besitzt und seine Funktion innerhalb eines geschlossenen Rahmens (wie dem Netzwerk einer Firma oder unserem Pubring) verbleiben muss.
Seine Auszeichnung (und damit seine Eigenschaften) durch unsere Signatur kann der Meta-Introducer quasi an weitere Personen, bzw. Keys (die Trusted Introducer) durch dessen Signierung "vererben", so dass auch alle Keys, die der Meta-Introducer signiert voll gültig (valid) und die Keybesitzer unser volles Vertrauen (trust) genies-sen, der Meta-Introducer "handelt" also quasi "stellvertetend" für uns selbst. Die Trusted-Introducer können diese Eigenschaft nicht weiter vererben, aber sich

stellvertretend für den Meta-Introducer für die Gültigkeit von Public Keys (Validity) verbürgen.
Im PGPkeys Fenster findet sich unter dem Key des Meta-Introducers als Ausweis unsere Signatur mit der Beschreibung: "RSA,DH/DSS meta-introducer signature", bei dem Trusted-Introducer "RSA,DH/DSS trusted-introducer signature".
Die Signatur eines Trusted-Introducers ist exportierbar, d. h. er kann auch ausserhalb unseres geschlossenen Rahmens für die Gültigkeit anderer Public Keys bürgen, allerdings kann man den Domainbereich einschränken, in dem er für Keys bürgen kann, in dem man unter "Domain restriction" die E-Mail Domains der Keys eingibt.
Erhalten wir jetzt aber einen Public Key, der von einem Trusted-Introducer signiert wurde, so ist der neue Key sogleich als gültig (valid) eingestuft, ohne dass wir den Key überprüft oder selbst signiert hätten, bzw. müssten.

Expiration (Verfallsdatum)

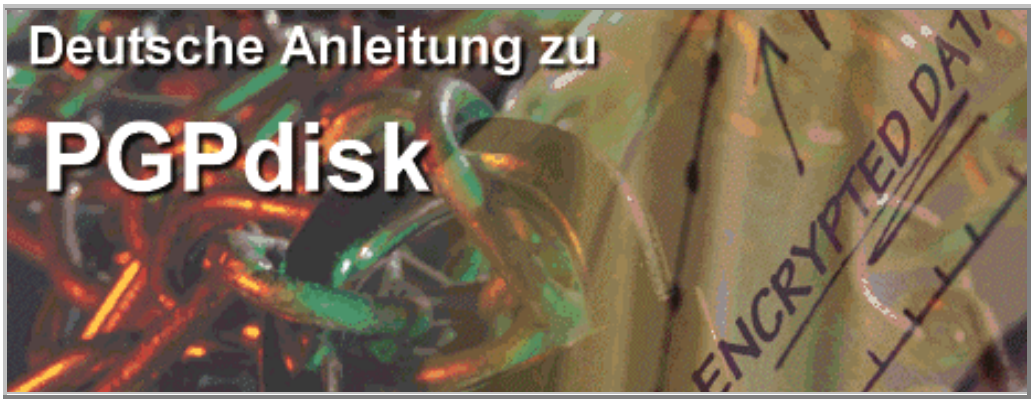
hier kann noch angegeben werden, ob unsere Signatur immer gültig (never) oder zu einem bestimmten Datum ungültig werden soll.

Hinweise

Zu den Begriffen Vertrauen in einen Keybesitzer (Trust) und Gültigkeit des Keys des Keybesitzers (Validity) siehe Web of Trust.
Es ist mit PGP 5/6 möglich, einen RSA Public Key mit einer DSS Signatur zu versehen. Aus Gründen der Kompatibilität sollte man aber einen RSA Public Key auch mit einem RSA Key signieren.
Nach dem Signieren kann der unterschriebene Key an den Besitzer und/oder die Keyserver versendet werden, je nach Typ wie in den Erklärungen der vorhergehenden Kapitel.
Das Konzept der Hierarchie von Meta-Introducer und Trusted-Introducer zielt auf Bereiche ab, in dem grosse Mengen an Public Keys und verschiedene Bereiche (Domains), aus dem diese Keys stammen, verwaltet werden müssen. So wäre es denkbar, dass der Personalleiter oder Prokurist einer Firma von der Geschäftsführung zum Meta-Introducer ernannt wird und der Personalleiter dann seine direkten Untergebenen, z. B. die Abteilungsleiter zu Trusted-Introducern macht. Oder der Hauptverkaufsleiter ernennt als Meta-Introducer die Gebietsverkaufsleiter zu Trusted-Introducern.
Für einen lokalen Pubring auf einem Einzelplatz-PC ist dieses Konzept überdimensioniert, da man ja selbst quasi der Meta-Introducer ist. Es kann aber sinnvoll sein, die Zertifizierungskeys von anerkannten oder bekannten Zertifizierungsstellen (Certification Authorities), deren Gültigkeit man nach den Prinzipien des Web of Trust überprüft hat und deren Zertifizierungstechnik man vertraut, mit einer Trusted-Introducer Signatur auszustatten, wodurch alle aktuellen und zukünftigen Public Keys, die von diesen Stellen zertifiziert wurden, sogleich gültig sind und von uns nicht mehr überprüft werden müssen.

Signieren eines Public Keys bei **PGP 2.6.3**:

```
PGP 2.6.3> pgp -ks User-ID-des-Keys Eigene-User-ID LW:\pubring.pgp
```



von Kai Raven
Version 1.0

Was ist PGPdisk

Mit PGPdisk kann innerhalb eines Verzeichnisses der Festplatte oder eines externen Speichermediums eine Datei angelegt werden, in der als "Container" Daten aller Art gespeichert und Programme installiert werden können.
Man kann eine PGPdisk auch in einer weiteren PGPdisk speichern, eine Komprimierung einer PGPdisk ist nicht möglich, aber die einzelnen Daten einer PGPdisk können komprimiert in ihr gespeichert werden. Alle Daten in der Datei sind so lange mit dem CAST Algorithmus verschlüsselt, so lange sie nicht benutzt werden.
Zur Benutzung der Containerdatei (PGPdisk volume) und mit ihr die darin gespeicherten Daten gibt der Besitzer seine Passphrase (Master Passphrase) ein und die Containerdatei wird als zusätzliches, virtuelles Laufwerk in den Verzeichnisbaum "eingehängt" (mounted). Danach kann das PGPdisk Laufwerk wie jedes andere Laufwerk auch benutzt und mit den Daten alle bekannten Dateioperationen wie kopieren, löschen und speichern durchgeführt werden, z. B. können auch weitere Verzeichnisse innerhalb des PGPdisk Laufwerks angelegt werden.
Nach der Benutzung wird die Containerdatei wieder aus dem Verzeichnisbaum ausgehängt (unmounted), ein Zugriff auf die Daten ist nur wieder nach Eingabe der Passphrase möglich. Um weiteren Personen Zugriff auf die PGPdisk zu ermöglichen, können bis zu sieben weitere Passphrases (Alternative Passphrases) vergeben und/oder beliebig viele Public Keys Passphrases hinzugefügt werden.
Eine PGPdisk Containerdatei selbst ist nicht vor einem löschenden Zugriff geschützt, d. h. jeder, der Zugriff auf die PGPdisk Containerdatei hat, kann sie auch löschen (es sei denn man benutzt ein Betriebssystem, mit dem man Zugriffsberechtigungen vergeben kann) ! Ausserdem gelten für PGPdisk die gleichen Sicherheitsvorkehrungen, die man auch bei PGP 5/6 zusätzlich treffen sollte.
Eine PGPdisk Datei kann per E-Mail an einen Mailpartner versendet oder auf einen externen Datenträger kopiert werden, nach Speicherung der PGPdisk Datei auf dem Zielrechner kann diese genauso aktiviert (mounted) und nach Eingabe der Passphrase benutzt werden wie auf dem Heimatrechner.

Wichtiger Hinweis:

Im Readme zur PGP 6.02 DH Desktop Security Version teilt NAI mit, dass es in PGPdisk Version 1.0, wie es in PGPdisk 1.0 und den PGP 6.0 Programmversionen enthalten ist, ein Sicherheitsleck gibt, die anderen Bestandteile der PGP Pakete sind nicht davon betroffen. Die PGPdisk Version in PGP 6.02 DH Desktop Security behebt das Sicherheitsleck. Der Nachteil besteht darin, dass man PGP 6.02 DH installieren muss, das eine Benutzung von RSA-Keys nur über die Microsoft Cryptographic API des 128-bit Securitypatches für den Internet Explorer zulässt und die zulässige Keygrösse auf 2048 bit Keys beschränkt ist.

Eine Erzeugung von RSA Keys ist mit der PGP Version 6.02 DH Desktop Security gar nicht möglich.
Um trotzdem volle RSA Funktionalität mit der PGP 6.02 DH Desktop Security Version zu erhalten, benötigt man ein RSA Add-on, das zur Zeit nicht "frei" erhältlich ist.
Wer PGPdisk einsetzen möchte, sollte deshalb eine PGP Version ab 6.0.2 Personal Privacy Desktop Security RSA einsetzen.
Mit dem neuen PGPdisk kommt ein Konvertierprogramm, dass automatisch vorhandene PGPdisks in das neue Format umwandelt - alternative Passphrases und Public Keys gehen dabei verloren, die Masterpassphrase und ADKs (!) nicht.

Von einer Benutzung von PGPdisk 1.0 muss abgeraten werden

Alternativen zu PGPdisk

- ScramDisk
- BestCrypt
- Blowfish Advanced

Installation und Starten von PGPdisk

PGPdisk ist Bestandteil der kommerziellen Versionen (Business Security, Personal & Desktop Security) ab PGP 6.0 und kann direkt bei der Installation als Option ausgewählt werden.
Der Aufruf von PGPdisk erfolgt entweder über das PGPtray Icon, Eintrag "Launch PGPdisk" oder über das Startmenü "PGPdisk"

Voreinstellungen

werden über den Button "Prefs" aufgerufen

Auto unmount on computer sleep (nicht bei Win NT)

bei Computern mit "Sleep Modus" werden alle aktivierten (mounted) PGPdisks automatisch deaktiviert (unmounted), wenn der Computer in diesen Betriebszustand übergeht.

Auto unmount after 1 - 999 minutes of inactivity

bleibt der Computer die Dauer der angegebenen Minuten inaktiv, werden alle aktivierten (mounted) PGPdisks automatisch deaktiviert (unmounted).

Unmount hotkey

nach Eingabe der gewählten Hotkeykonstellation werden alle aktivierten (mounted) PGPdisks automatisch deaktiviert (unmounted).

Prevent sleep if any PGPdisk could not be unmounted (nicht bei Win NT)

wenn PGPdisks nicht deaktiviert (unmounted) werden können, geht der Computer auch nicht in den "Sleep Modus".

Hinweise:

- alle Optionen können zugleich konfiguriert werden

- die automatischen Deaktivierungsfunktionen können nicht ausgeführt werden, wenn und so lange Daten einer PGPdisk in Benutzung sind

Der Umgang mit einer PGPdisk

Eine PGPdisk herstellen

- PGPdisk aufrufen
- Button "New" anklicken, wodurch der "New PGPdisk wizard" gestartet wird
- Button "Weiter"
- Dateiname vergeben und Verzeichnis der Speicherung auswählen
- unter "PGPdisk Size" die Grösse der PGPdisk in Kilobyte, Megabyte oder Gigabyte angeben (der aktuell freie Speicherplatz ist im Hinweistext oberhalb angegeben)
- über "PGPdisk Drive Letter" den Laufwerksbuchstaben vergeben, unter dem die PGPdisk später in den Verzeichnisbaum eingehängt werden soll
- Button "Weiter"
- Passphrase mit einer Mindestzeichenlänge von 8 Zeichen eingeben, ist die Option "Hide Typing" aktiviert, wird die eingetippte Passphrase nicht angezeigt
- um Zufallsdaten zur Verschlüsselung der PGPdisk zu erzeugen, die Maus so lange klickend über das Fenster bewegen, bis die Balkenanzeige auf 100% steht
- 2x Button "Weiter"
- die PGPdisk wurde erstellt und unter dem Laufwerksbuchstaben eingehängt
- die PGPdisk muss anschliessend mit FAT/NTFS formatiert werden
- PGPdisk einsatzbereit

  Heinweis:
  Wenn ein PGPdisk Volume unter Windows 98 erstellt wurde, das sofort an einen anderen Ort kopiert werden soll, muss Windows 98 zuerst rebootet werden.

Eine PGPdisk mounten

- alle geöffneten Dateien oder Programme der PGPdisk schliessen
- PGPdisk aufrufen
- Button "Mount" anklicken
- Passphrase eingeben
- Laufwerksbuchstaben übernehmen oder aus der Liste auswählen
- Checkbox "Read-only" aktivieren, wenn ein nur lesender Zugriff ermöglich werden soll, bzw. wenn andere Benutzer des Computers (lesenden) Zugriff auf die Daten der PGPdisk erhalten sollen
- OK

oder direkt im Explorer die PGPdisk Datei öffnen und nach obiger Vorgehensweise vorgehen

Eine PGPdisk unmounten

- PGPdisk aufrufen
- Button Unmount anklicken oder
- im Explorer auf das Laufwerksicon klicken und im Kontextmenü über PGPdisk
- Unmount PGPdisk anklicken

Passphrase- und Public Key Management

Alternative Passphrases

Einer PGPdisk können bis zu sieben weitere, alternative Passphrases zugeordnet werden, um sieben weiteren Personen den Zugriff zu ermöglichen, jedoch kann nur der Inhaber der Master Passphrase alternative Passphrases anlegen. Der Inhaber der alternativen Passphrase kann seine Passphrase ändern.

Vorgehensweise:

- betreffende PGPdisk deaktivieren, sollte sie gemountet sein
- PGPdisk aufrufen oder im Explorer den PGPdisk Eintrag des Kontextmenüs der PGPdisk Datei
- "Add Passphrase" über das "File" Menü auswählen
- PGPdisk Containerdatei öffnen
- Master Passphrase eingeben
- 2 x die alternative Passphrase eingeben
- Checkbox "Read-only" aktivieren, wenn der Benutzer mit Passphrase XY einen nur lesenden Zugriff auf die PGPdisk bekommen soll

Passphrase ändern

- betreffende PGPdisk deaktivieren, sollte sie gemountet sein
- PGPdisk aufrufen oder im Explorer den PGPdisk Eintrag des Kontextmenüs der PGPdisk Datei
- "Change Passphrase" über das "File" Menü auswählen
- PGPdisk Containerdatei öffnen
- alte Master Passphrase oder Alternative Passphrase eingeben
- neue Master Passphrase oder Alternative Passphrase eingeben

Alternative Passphrase löschen

- betreffende PGPdisk deaktivieren, sollte sie gemountet sein
- PGPdisk aufrufen oder im Explorer den PGPdisk Eintrag des Kontextmenüs der PGPdisk Datei
- "Remove Passphrase" über das "File" Menü auswählen
- PGPdisk Containerdatei öffnen
- zu löschende Alternative Passphrase eingeben

Alle alternativen Passphrases löschen

- betreffende PGPdisk deaktivieren, sollte sie gemountet sein
- PGPdisk aufrufen
- Shift-Key drücken und gedrückt halten
- "Remove Passphrase" über das "File" Menü auswählen
- Abfragedialog bestätigen oder
- Explorer aufrufen und Verzeichnis mit PGPdisk öffnen
- Shift-Key drücken und gedrückt halten
- "Remove Passphrase" über den PGPdisk Eintrag des Kontextmenüs der PGPdisk auswählen
- Abfragedialog bestätigen

Public Key Passphrases zuordnen

Vorgehensweise:

- betreffende PGPdisk deaktivieren, sollte sie gemountet sein
- PGPdisk aufrufen oder im Explorer den PGPdisk Eintrag des Kontextmenüs der PGPdisk Datei
- "Add Public Keys" über das "File" Menü auswählen
- PGPdisk Containerdatei öffnen
- Master Passphrase eingeben
- im Recipient Selection Dialog Fenster alle gewünschten Public Keys in das untere Fenster ziehen
- Checkbox "Read-only" aktivieren, wenn der Benutzer mit Passphrase XY einen nur lesenden Zugriff auf die PGPdisk bekommen soll
- OK

Public Keys Passphrases löschen

- betreffende PGPdisk deaktivieren, sollte sie gemountet sein
- PGPdisk aufrufen oder im Explorer den PGPdisk Eintrag des Kontextmenüs der PGPdisk Datei
- "Remove Public Keys" über das "File" Menü auswählen
- PGPdisk Containerdatei öffnen
- Master Passphrase eingeben
- im Recipient Selection Dialog Fenster alle gewünschten Public Keys im unteren Fenster in das obere Fenster ziehen
- OK

Hinweise:
Wenn ein Split Key einer PGPdisk hinzugefügt wird, muss vor der Benutzung des Split Keys zur Aktivierung der PGPdisk der Split Key zuerst über PGPkeys wieder zusammengefügt werden.
Die Benutzung eines Split Keys ist also im Zusammenhang mit PGPdisk nicht praktikabel.
Wenn PGPdisk Teil einer PGP 6 Clientversion ist, die von einem Administrator für die Benutzer erstellt wurde, kann dieser wie für das übrige PGP Programm auch für PGPdisk einen ADK/CMRK-Key bestimmen. Nähere Informationen zu ADK/CMRK unter Abschaffung der "Privacy" bei PGP 5/6 und die Diskussion um GAK, CMRK, ARR, MRK

PGP 6.5.X oder
"Die neuen Funktionen der PGP 6.5.X Versionen"

Dieser Text geht vom Einsatz von PGP ab Version 6.5.1 Personal Privacy aus.

Erstellung von selbstentschlüsselnden Archiven (SDA)

Neben den bekannten Verschlüsselungswegen ist mit PGP 6.5.1 eine neue Verschlüsselungsform eingeführt worden.
Über die SDA´s werden einzelne Dateien oder Verzeichnisse nach Eingabe eines Passwortes konventionell (also ohne den Public Key des Empfängers) verschlüsselt und komprimiert. Die so behandelten Daten liegen danach als ausführbare Archivdatei vor, deren Inhalte (Dateien und Verzeichnisse) nach Eingabe des Passwortes entschlüsselt und dekomprimiert werden. Dieser Verschlüsselungsweg eignet sich zur Kommunikation mit Empfängern, die selbst kein PGP einsetzen. Natürlich setzt der Gebrauch von SDA´s voraus, dass das Passwort vorher über einen "sicheren Kanal" ausgetauscht wurde. Welcher Algorithmus dabei zum Einsatz kommt, ist in der Dokumentation nicht angegeben.
Im Manual zur MacOS Version heisst es aber:

"The self-decrypting archives created on a Mac with work on either PowerPC and 68K Macs, and are encrypted using the CAST algorithm."

Anmerkung:
Man sollte sich darüber klar sein, dass nicht jedes PGP Archiv, das man als E-Mail Attachement erhält, wirklich ein PGP Archiv ist, sondern auch einen Virus enthalten kann: Eine Datei im verschlüsselten Archiv kann mit einem Virus infiziert sein, das SDA als ausführbare Datei kann selbst infiziert sein oder über die Extraktion wird ein ausführbares Virus auf dem Computer installiert. Deshalb erscheint es ratsam, vor und nach dem Öffnen zuerst auf Viren zu scannen, oder ein Attachement, das als PGP Archiv daherkommt, aber aus unsicherer, bzw. unbekannter Quelle stammt, erst gar nicht zu öffnen.

Vorgehensweise:

1. über die PGPtools oder das Explorerkontextmenü PGP die Funktion "encrypt" aufrufen
2. Checkbox "Self Decrypting Archive" aktivieren (dabei wird automatisch auch die Checkbox Conventional Encryption aktiviert und das Empfängerauswahlfenster abgeblendet)
3. Nach Bestätigung mit "OK" 2x das gewählte Passwort eingeben

Hot Keys

In den Optionen können Hotkeykombinationen für die Funktion Use current window (die Inhalte des aktiven Fensters eines Programmes werden entschlüsselt, verschlüsselt und/oder signiert) festgelegt werden.

Die gleichen Funktionen sind nach wie vor über PGPtray zu erreichen.
Bevorzugt man das Caching der Passphrase über einen längeren Zeitraum, kann man auch eine Hotkeykombination für das Leeren des Passphrase-Speichers eingeben.

PGPtray Options/Hotkeys:

Purge passhrase caches	Passphrase Speicher leeren
Encrypt current window	Inhalt des aktiven Fensters verschlüsseln
Sign current window	Inhalt des aktiven Fensters signieren
Encrypt & Sign current window	Inhalt des aktiven Fensters verschlüsseln & signieren
Decrypt & Verify current window	Inhalt des aktiven Fensters entschlüsseln und überprüfen

Biometrische Fingerprint Wortliste

Der Fingerprint eines Public Keys kann als Hexadezimalwert oder als Wortliste dargestellt werden. Phil Zimmermann hat dafür den Begriff "Biometrische Wortliste" gewählt, in Anlehnung an den Gebrauch von Wörtern im militärischen Bereich zur Übermittlung von Informationen und der biometrischen Identifizierungmethoden wie Irisscan, Fingerabdruck, Stimmenprofil und Gesichtsgeometrie. Der Zweck liegt darin, die Fehleranfälligkeit bei dem Austausch der Fingerprints durch Vorsagen der Hexadezimalzahlen zu minimieren, da Wörter eindeutiger seien als eine alphanumerische Reihe. Dementsprechend gibt es 256 verschiedene Wörter für 255 verschiedene Bytekombinationen.
Die Anzeige der "Biometrischen" Fingerprintwortliste erreicht man über den Aufruf der Key Properties.

Löschung des freien Festplattenplatzes per Zeitplan

Wie bisher kann man Daten mittels PGP über das Explorerkontextmenü und über die PGPtools löschen. Neu hinzugekommen ist die Möglichkeit, das Löschen der freien Bereiche einer Fetsplatte oder Partition über den Taskplaner zu automatisieren.

Vorgehensweise:

1. Aufruf der PGPtools
2. "Freespace Wipe" anklicken
3. in der Laufliste das Laufwerk ("Wipe drive:") und die Anzahl der Überschreibungsrunden ("passes") angeben
4. Button "Schedule" anklicken oder
   Button "Beginn Wipe", wenn eine einmalige Löschung vorgenommen werden soll
5. den Zeitplan nach eigenen Bedürfnissen konfigurieren

Anmerkung:

PGP erlaubt bis zu 32 Runden, d. h. der freie Bereich wird 32 x überschrieben. Laut PGP sei bekannt, dass Firmen, die sich mit der Wiederherstellung von Daten beschäftigen, Daten rekonstruieren können, die bis zu 9x überschrieben wurden, so dass man hier einen Wert > 9 eintragen sollte.
Während des Löschens und Überschreibens müssen alle anderen Applikationen geschlossen werden, da der Free Space Wipeprozess durch andere Schreibprozess anderer Applikationen zurückgesetzt wird.
Um eine 100% Löschung des Datenbestandes zu erreichen, z. B. wenn die Festplatte abgegeben wird, sollte man die grösste Rundenzahl wählen, zusätzlich ein weiteres Löschprogramm anwenden und abschliessend die Einträge der Dateinamen löschen.

Soll die Festplatte entsorgt werden, ist ebenfalls so zu verfahren, zusätzlich kann man die Festplatte noch mit einem starken Magneten behandeln, zerkratzen, einstampfen, kleinraspeln und einschmelzen – und währenddessen einen bewaffneten Sicherheitsdienst Wache schieben lassen (wie es die NSA mit ihren Festplatten macht) ;-)

Virtual Private Network (VPN)

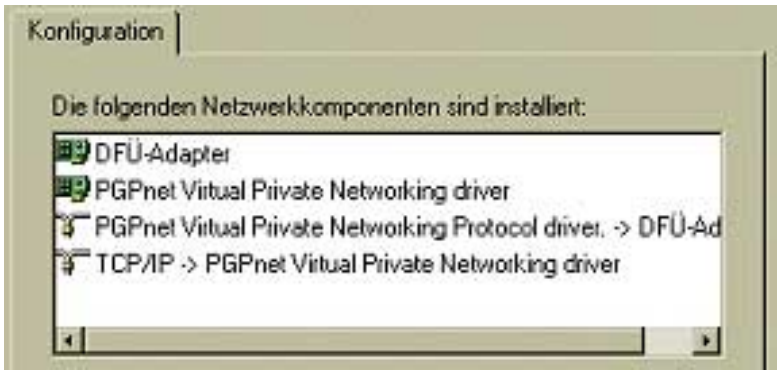
Ein virtuelles privates Netzwerk ermöglicht einem Benutzer von einer privaten Lokalität über das Internet eine sichere Verbindung (*"Secure Association"* SA) zu anderen Benutzern oder Firmen und Organisationen in der Art herzustellen, als würde sich der Benutzer im Intranet einer Firma befinden oder er und ein weiterer Benutzer hätten ein eigenes Intranet aufgebaut und gehörten zum gleichen Netzwerk. Das Internet wird quasi zur Auslagerung des firmeninternen Netzwerkes, man könnte auch sagen, im Netz des Internets als Transportmedium wird für diese Verbindung ein zweites, temporäres Netz angelegt, in dem die zwei Rechner zweier Benutzer, der Rechner eines Benutzer und der Rechner eines Intranetzes oder die Rechner zweier verschiedener Intranets kommunizieren. Das Besondere am VPN ist, dass für diese Verbindung zwei spezielle Protokolle eingesetzt werden.
Das erste Protokoll heisst "IPsec (Internet Protokol Security)-Protokoll", das sich wiederum aus dem Encapsulating Security Payload (ESP) und Authenticated Header (AH) Protokoll zusammensetzt.
Der sogenannte Authentication Header (AH) definiert dabei den Hash (kryptografisch gebildete Prüfsumme)des gesammten IP-Paketes zur Itegritätskontrolle, während der Encapsulating Security Payload Header (ESP) die Art und Weise definiert, wie die eigentlichen Daten verschlüsselt zu übertragen sind.
ESP erlaubt den Tunnel Mode und den Transport Mode. Beim Tunnel Mode wird das gesamte IP-Paket inklusive IP-Header des eigentlichen Zielrechners verschlüsselt dem ESP-Header angehängt, der äussere Header trägt dabei IP-Adresse eines Gatewayrechners, so dass die IP Pakete durch den Gateway zum Zielrechner wie durch einen verschlüsselten Tunnel übertragen werden. Beim Transport Mode werden nur TCP,UDP und ICMP verschlüsselt dem Header angehängt.
Das zweite Protokoll heisst "Internet Key Exchange (IKE) Protokoll", in dem festgelegt wird, welche Algorithmen und Schlüssel zur wechselseitigen Authentifizierungs- und Schlüsselaustauschphase und zur anschliessenden Verschlüsselung der zu übertragenden Nutzungsdaten verwendet werden.
"PGPnet" als VPN basiert auf dem IPsec- und IKE-Protokoll und stellt gelichzeitig unter der gleichen Bezeichnung "PGPnet" die Applikation zur Verwaltung der VPNs dar. Voraussetzung zur Nutzung von PGPnet für Einzelbenutzer mit Einwahlverbindung und dynamischer IP-Adresse ist die Kenntnis der eigenen wie auch der IP-Adresse des Zielnetzes, Gateways oder Kommunikationspartners.
PGPnet arbeitet mit Gauntlet VPN, Cisco Routern (ab Cisco IOS 12.0(4) mit IPSec TripleDES) und Linux FreeS/WAN zusammen und unterstützt die Verwendung von OpenPGP Keys und X.509 Zertifikaten zur Authentifizierung.

PGPnet oder
"Die Konfiguration und Anwendung des VPN PGPnet"

Dieser Text geht vom Einsatz von PGP ab Version 6.5.1 Personal Privacy aus.

Installation

Bei der Installation von PGP 6.5.X kann die Option PGPnet aktiviert werden. Ist die Gesamtinstallation abgeschlossen, fragt PGP, welches Netzwerkinterface an PGPnet gebunden werden soll, bzw. zeigt das aktuell zu sichernde Netzwerkinterface an. Das kann der DFÜ-Adapter (Modem, ISDN-Karte), die Ethernetkarte oder ein RAS-WAN-Adapter sein.



In der Netzwerkkonfiguration der Systemsteuerung sieht man den PGPnet VPN Treiber, an den das TCP/IP Protokoll gebunden ist, während der DFÜ-Adapater an das PGPnet VPN Protokoll gebunden wurde.

Wenn die Netzwerkkonfiguration geändert, ein zusätzliches Netzwerkinterface eingebaut oder das aktuelle Netzwerkinterface ausgetauscht wurde, muss man anschliessend über Set Adapter im PGP Startmenü das Netzwerkinterface bestimmen und einen Neustart durchführen.

Ports

Bei parallelem Einsatz einer Firewall/Proxy müssen folgende Ports geöffnet sein:

- 500 UDP
- 51 TCP
- 50 TCP

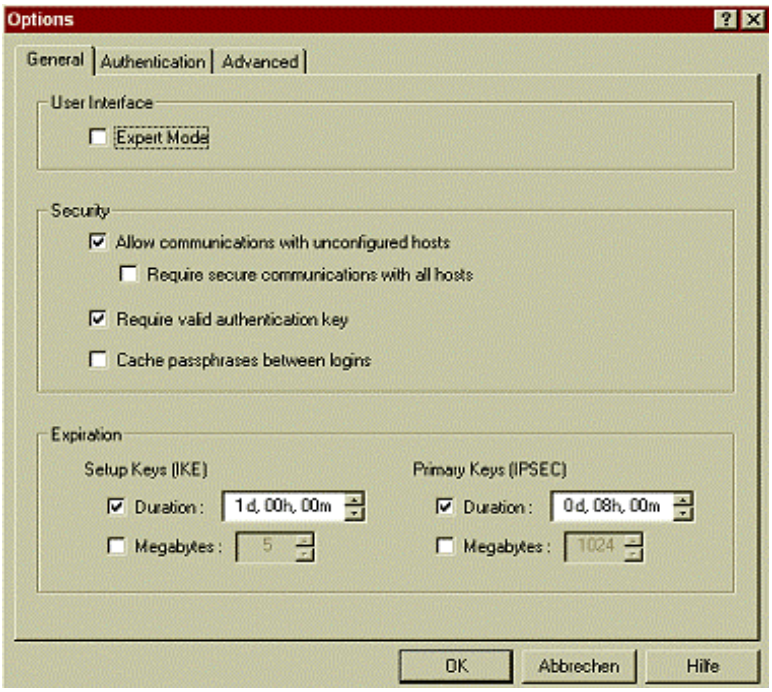
Deinstallation

PGPnet sollte nicht manuell über das Löschen einzelner PGPnet Dateien oder der Entfernung von Einträgen in den Netzwerkoptionen durchgeführt werden. Es ist notwendig, PGP insgesamt über die Deinstallierroutinen zu entfernen.

Konfiguration

Aufruf über PGP Systemtray/PGPnet/Options

General



User Interface

Expert Mode
der Neueintrag eines Rechners wird manuell durchgeführt, der Konfigurationsassistent wird nicht benutzt

Security

Allow communications with unconfigured hosts
die Verbindung zu Rechnern, die nicht in PGPnet eingetragen sind, ist immer möglich Sollte aktiviert sein, wenn man sich im WWW zu einer Vielzahl von Rechnern verbindet, die wenigen Rechner, zu denen man eine gesicherte PGPnet Verbindung aufbauen möchte, werden dann in die Hostliste eingetragen.

Require secure communications with all hosts
PGPnet versucht automatisch eine sichere Verbindung zum Kontaktrechner aufzubauen. Ist der Kontaktrechner nicht mit PGPnet ausgestattet, kommt keine Verbindung zustande, es sei denn, er wurde als unsicherer Rechner in die Hostliste eingetragen. Diese Option ist sinnvoll, wenn sich der eigene Rechner in einer Umgebung befindet, in der die meisten Zielrechner ebenfalls mit PGPnet ausgestattet sind oder wenn sichere PGPnet Verbindungen bevorzugt werden.

Require valid authentication key
PGPnet akzeptiert Keys von anderen Rechnern nur dann, wenn sie im lokalen Keyring als valid (gültig) gekennzeichnet sind. Auf der Cient Seite (z. B. dem heimischen Rechner) sollte die Option aktiviert werden, dazu müssen dann alle Keys der Rechner und Server, zu denen

eine sichere PGPnet Verbindung aufgebaut werden soll, in den Keyring aufgenommen und vom Benutzer verifiziert (d. h. nach den Regeln des Web of Trust überprüft und signiert) werden. Auf einem Server sollte die Option deaktiv bleiben, da es zu einer Vielzahl von Verbindungen kommt und nicht alle PGP Keys der Anfragenden verwaltet und verifiziert werden können.

Cache passphrases between logins
Vor der Verwendung von PGPnet muss sich der Benutzer in PGPnet "einloggen". Wird diese Option aktiviert, speichert PGPnet die Passphrase während einer Windows-Session. Erst bei einem Neustart oder wenn man den Authentifizierungskey löscht, wird der Passphrasespeicher geleert.

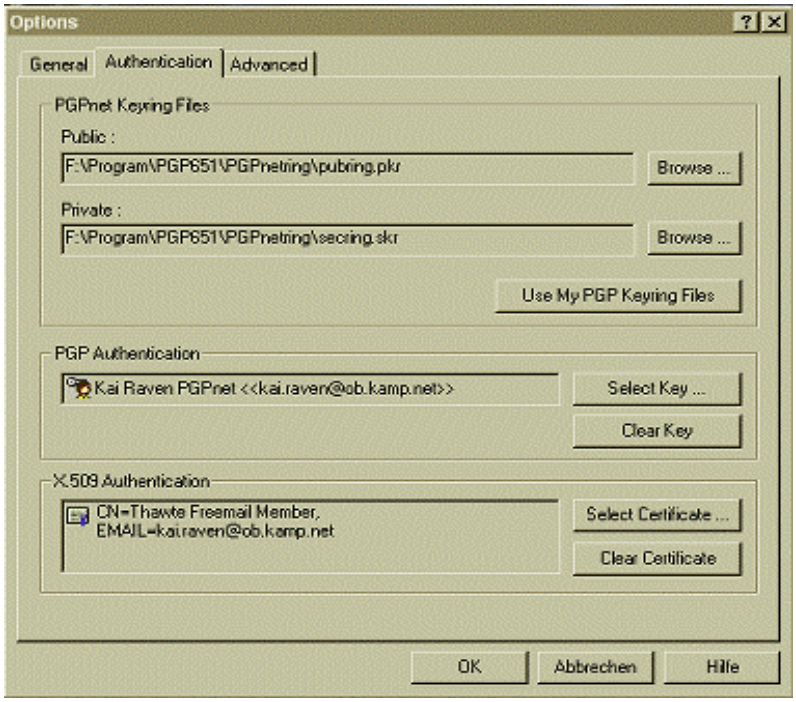
Expiration

Hier wird festgelegt, wie lange die Keys, die zum Aushandeln der Schlüsselaustauschmodalitäten (IKE-Keys) und die Keys, die zur Verschlüsselung und Authentifizierung (IPsec-Keys) und damit der Secure Associations, gültig sind. Dabei wird die restriktivere Einstellung einer der Kommunikationspartner vorrangig gültig, d. h. hat man selbst die Dauer auf einen Tag festgelegt, der Kommunikationspartner aber auf 30 Minuten, sind die Keys auch nur 30 Minuten gültig. Erreichen die Keys, bzw. eine SA ihren Verfallszeitpunkt, baut PGPnet automatisch eine neue SA auf. Der Gültigkeitsstatus ist aus dem Statusfenster von PGPnet ersichtlich.

Duration
Zeitdauer der Gültigkeit in X d (Tagen) Y h (Stunden) Z m (Minuten)

Megabytes
Zeitdauer (eher gesagt Mengendauer) der Gültigkeit in Megabytes an Daten, die während einer SA übertragen wurden, d. h. wenn bei einem Wert von 5 MB während einer SA, die nach Zeit 5 Tage gültig ist, nach 2 Tagen 5 MB übertragen wurden, verfällt die SA.

Authentication



PGPnet Keyring Files

Über diesen Dialog kann eine spezielle Public und Secret Keyringdatei für PGPnet bestimmt werden, die den Public und Secret Key enthalten, der PGPnet zur Authentifizierung des eigenen Rechners gegenüber dem Kontaktrechner dient. Über den Button *Use My PGP Keyring Files* wird der bestehende PGP Public und Secret Keyring in PGPnet Keyring Files eingetragen.

PGP Authentication

Will man PGP Keys zur Authentifizierung einsetzen, wird über den Button *Select Key* der PGP aus den PGPnet Keyringfiles bestimmt, der zur Authentifizierung dienen soll. Nach Auswahl des Keys muss die entsprechende Passphrase des Keys eingegeben werden. Die gleiche Passphrase wird jedesmal beim Login in PGPnet angefragt.
Wichtig: Beide Kommunikationspartner müssen hier den gleichen Schlüsseltyp angeben und verwenden, also RSA oder DH.
Sollte sich das Problem einstellen, dass man zwar die schon vorhandenen Keyringdateien in PGPnet angegeben hat, aber daraus den vorhanden Public PGP Key nicht zum Authentifizierungs-Key bestimmen kann, bietet sich folgende Lösung an:

a. mit PGPkeys einen neuen, nur für PGPnet vorgesehenen PGP-Key erzeugen und diesen mit dem schon vorhandenen, eigenen PGP-Key signieren
b. die Keyringe auf der Festplatte kopieren, bzw. duplizieren und die Pfadangaben in den PGP Optionen auf die neuen PGPnet Keyringe setzen
c. alle Keys bis auf den neuen PGPnet-Key und den PGP-Key löschen
d. danach PGP wieder auf die eigentlichen Keyringe setzten und eventuell daraus den PGPnet-Key entfernen
e. in den Optionen von PGPnet die Pfadangeben zu den Keyringen zuerst auf die ursprünglichen Keyringe setzen und anschliessend auf die neu erstellten PGPnet Key-

ringe.
PGPnet benutzt jetzt eigene Keyringe, während PGP die alten Keyringe benutzt.

X.509 Authentication

zur Authentifizierung wird ein X.509 Zertifikat, wie es von der S/MIME Verschlüsselung und Signierung mittels S/MIME kompatiblen Mailsystemen wie Outlook oder dem Netscape Messenger bekannt ist, benutzt. Zusätzlich zur Passphrase des PGP-Keys wird beim Login in PGPnet die Passphrase des Zertifikats abgefragt.

Wichtig: Beide Kommunikationspartner müssen die gleiche Root CA verwenden und das gleiche, von ihnen signierte und mit grösstem Trustlevel versehene Root CA Zertifikat in ihrem PGPnet Pubring haben(siehe unten).

X.509 Zertifikate

Zertifikate sind digitale Dokumente, die die Bindung eines Public Keys an eine Person,Organisation oder Firma attestieren, so dass man über das Zertifikat prüfen kann, ob ein Public Key zu dieser Einheit gehört.
Das Zertifikat besteht aus dem Public und Secret Key, der Signatur des Inhabers und der Signatur der Zertifizierungsstelle. Darüber hinaus kann ein Zertifikat die X.509 Version (aktuell X.509 v3), eine Seriennummer, Gültigkeitsdauer und weitere Datenfelder enthalten.
Je nach Grad der Überprüfung des Eigentums einer Einheit an einem Public Key kann man verschiedene Zertifikatsklassen unterscheiden. So wird z. B. bei Class 1 Zertifikaten die Prüfung über die Existenz einer E-Mailadresse und verschiedener Rückmeldungen des E-Mailadresseninhabers an die CA durchgeführt. Bei Class 2 Zertifikaten kommt z. B. noch das Post-Ident Verfahren dazu und bei Class 3 Zertifikaten muss der Keyinhaber persönlich bei der CA erscheinen oder zusätzliche Beweise seiner Identität und Eignerschaft beibringen. Das Zertifikat in der obigen Abbildung ist z. B. ein Class 1 Zertifikat.
X.509 Zertifikate sind kompatibel zum Public-Key Cryptography Standard (PKCS) 6, der von den RSA Laboratories in Kooperation mit Apple, Microsoft, DEC, Lotus, Sun und MIT entworfen wurde. Das bekannteste Zertifikatsformat ist im ITU-T X.509 Standard definiert, daher der Name X.509 Zertifikat. Die X.509 Zertifikate verzichten ganz auf das Web of Trust Modell von PGP und setzen statt dessen auf Authentifizierung über eine hierarchische CA Struktur.

Der Vorgang verkürzt:

Um die X.509 Authentifizierung zu benutzen, muss zunächst das X.509 Zertifikat der Root CA (Root Certification Authority - Wurzel- oder Haupt-Zertifizierungsstelle) über einen Webbrowser heruntergeladen und in den Pubring importiert, die Information zur Root CA in den CA Optionen von PGP eingeben, eine Zertifikatsanfrage für den PGP Key an die Root CA über PGP abgeben, das erhaltene Zertifikat dem PGP-Key zugeordnet und abschliessend in PGPnet das erhaltene Zertifikat zum Authentifizierungs-Zertifikat bestimmt werden.

Der Vorgang in Einzelschritten:

(bei Verwendung eines Net Tools PKI Servers, was immer das auch sein mag)

1. **Das Root CA Zertifikat in den Pubring integrieren**
   a. mit einem Webbrowser eine der Zertifizierungsstellen im WWW aufsuchen.

b. auf der Seite, auf der die CA Zertifikate zum Download stehen, das Root CA Zertifikat anklicken
c. die Detailinformationen des Zertifikats nach Beendigung des Ladens aufrufen und den Public Keyblock des Root CA Zertifikats kopieren und in PGPkeys über den Keyimportdialog in den Pubring importieren.
d. das Root CA Zertifikat mit dem eigenen (PGPnet-) PGP-Key signieren und in den Key Properties den Trustlevel auf Maximum stellen.

2. **Die CA Optionen festlegen**
   a. PGP Optionen im PGPtray aufrufen und Karteireiter CA auswählen
   b. die URL im Textfeld *Certificate Authority* eingeben, über die man das Root CA Zertifikat erhalten hat
   c. die URL im Textfeld *Revocation URL* eingeben, über die man die Certificate Revocation List (CRL) erhält, das sind Listen, in denen die Informationen über zurückgezogene Zertifikate gespeichert werden
   d. in der Type Liste den Namen der CA angeben, die man benutzt:
      - NAI Net Tools PKI Server
      - VeriSign OnSite (60 Tage Probezertifikat, sonst ca. 10 $/Jahr)
      - Entrust (60 Tage Probezertifikate)

      Weitere CA´s, die X.509 Zertifikate ausstellen:

      - Thawte (kostenlose Zertifikate mit Angabe von Adresse und Personalausweisnummer (die nicht überprüft wird)
      - TC TrustCenter (kostenlose Class 1,2 und 3 Zertifikate)
   e. über den Button *Select Certificate* das Root CA Zertifikat auswählen.
      Im Textfeld darunter erscheinen nun nähere Informationen zum Root CA Zertifikat, die von CA zu CA variieren können, da die Angaben abhängig von den Regelungen der CA sind:

CN	Common Name	Beschreibung des Zertifikatstyps, z. B. "Root"
EMAIL		die E-Mailadresse des Zertifikatseigentümers
OU	Organizational Unit	die Abteilung der Organisation, zu der das Zertifikat gehört
O	Organization name	der Name des Unternehmens, zu der das Zertifikat gehört
L	Locality	z. B. die Stadt, in der der Zertifikatsinhaber seinen Sitz hat
C	Country	der Staat, in dem der Zertifikatsinhaber seinen Sitz hat, z.B. "Germany"
ST	State	z. B. der Bundesstaat oder das Bundesland, in dem der Zertifikatsinhaber seinen Sitz hat

3. **das eigene Zertifikat anfordern**
   a. in PGPkeys den gewünschten PGP-Key markieren und im Kontextmenü Add/Certificate anwählen
   b. im Dialogfenster Certificate Attributes können nun die Angaben ausgewählt werden, die später im Zertifikat enthalten sein sollen (siehe oben)
   c. im PGP Passphrase Fenster die Passphrase des PGP-Keys eingeben

d. PGP verbindet sich jetzt mit dem CA Server, der in den CA Optionen unter Punkt 2 angegeben wurde und sendet die Anfrage, nachdem sich der CA Server gegenüber dem eigenen Rechner ausgewiesen hat

4. **das eigene Zertifikat in den Pubring aufnehmen**
    a. nach Absendung der Zertifikatsanforderung erhält man eine Nachricht, dass das Zertifikat heruntergeladen werden kann
    b. in PGPkeys den entsprechenden PGP-Key markieren
    c. im Menü Server den Menüpunkt Retrieve Certificate anklicken, worauf sich PGP erneut mit dem CA Server verbindet, das Zertifikat herunterlädt und in den Pubring importiert
    d. die PGPnet Optionen aufrufen und im Karteireiter Authentification unter X.509 Authentification über den Button Select Certificate (siehe Abbildung oben) das erhaltene Zertifikat auswählen

Probleme

Wie aus der Auflistung der CAs ersichtlich, bieten die zwei CAs, die PGP vorschlägt nur 60 Tage Probezertifikate an, deshalb wurden sie von mir nicht getestet. Bei Thawte und TC TrustCenter kann man nicht den oben beschriebenen Weg gehen, da die beiden CA´s inkompatibel zum PGP System der Zertifikatsanforderung sind. So muss man auf deren Webseiten zusätzliche Angaben und Passwörter angeben, ohne die ein Antrag erst gar nicht entgegengenommen wird. Die zusätzlichen Prüfverfahren über Rücksendung einer Prüf E-Mail oder das Post-Ident Verfahren verhindern einen automatischen Zertifikatsimport, wie PGP ihn vorschlägt.

Was bleibt ?

Man muss zuerst bei Thawte oder TC Trustcenter auf herkömmlichen Wege, d. h. über deren Webseiten ein Zertifikat beantragen und anfordern und anschliessend das Zertifikat manuell in den Pubring "einpflegen"

Vorgehensweise

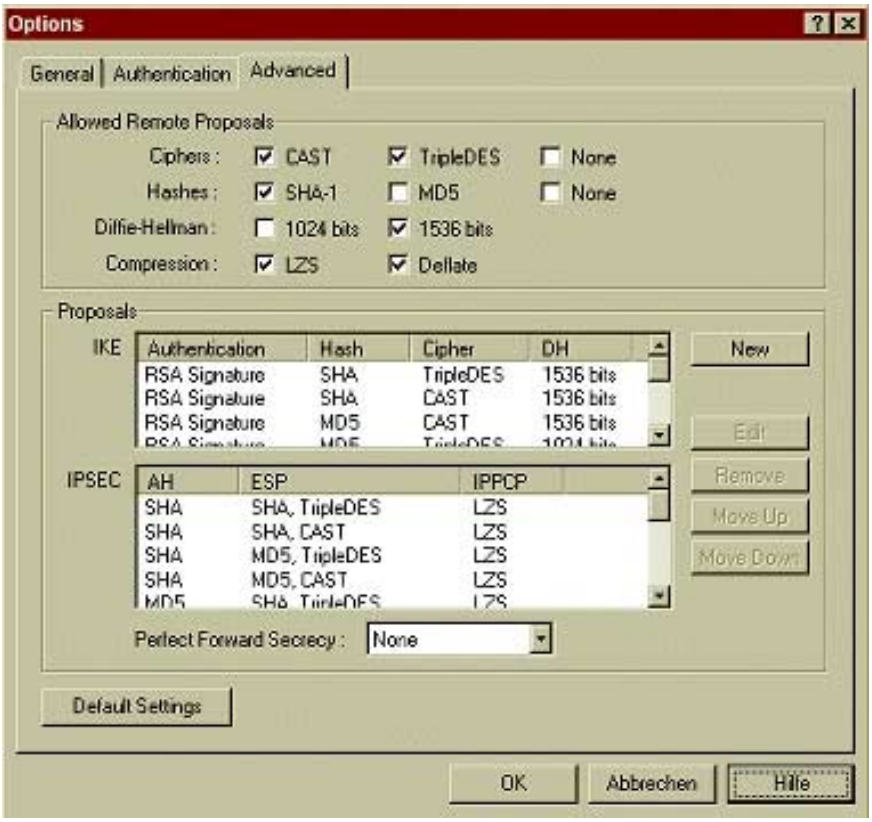
1. das Root CA Zertifikat der Zertifizierungsinstitution in den Web Browser installieren, dazu auf die Zertifikatsangebotsseite der CA gehen und die entsprechenden Links anklicken
2. die Browser starten daraufhin einen Aufnahmevorgang, in dem man durch die Installationsprozedur geführt wird
3. das Root CA Zertifikat anhand Fingerprints/Anruf bei der CA überprüfen
4. über eine 128-bit SSL Verbindung auf der Webseite, auf der Angaben zur Identität und zur Zertifikatsanforderung gemacht werden, das Zertifikat anfordern
5. nach Durchlaufen der Identitätsfeststellungsverfahren wie Post-Ident oder Rücksendung einer Authentizitätsnachricht per E-Mail und dem Erhalt der E-Mailbestätigung der CA, dass das Zertifikat zum Download bereitliegt, mit dem Web-Browser die angegebene Webseite aufsuchen und das persönliche Zertifikat in den Browser installieren
6. das persönliche Zertifikat und das Root CA Zertifikat aus dem Browser als Datei exportieren.
   Dabei muss man beachten, dass das Zertifikat als PKCS-12 Zertifikat (Endung:pfx oder p12) oder als PEM (Internet Privacy-Enhanced Mail Standard, der wie PKCS der gesicherten elektronischen Datenübertragung dient) Zertifikat (Endung: pem) exportiert wird, da PGP nur Zertifikate, die in dieser Form vorliegen, importieren kann.

Desweiteren, dass sowohl Public als auch Secret Key des persönlichen Zertifikats und alle beteiligten Zertifikate mitexportiert werden

7. PGPkeys aufrufen und aus dem Explorer mit der Maus die exportierten PEM, PFX oder P12 Zertifikatdateien in das PGPkeys Fenster ziehen oder über das Menü Keys/Import das persönliche Zertifikat nach Eingabe der Passphrase und das Root CA Zertifikat in den Pubring importieren
8. anschliessend den PGPnet PGP-Key mit dem persönlichen Zertifikat und das persönliche Zertifikat mit dem PGPnet PGP-Key signieren. Danach den Trustparameter des persönlichen Zertifikats und des Root CA Zertifikats auf Maximum stellen
9. die PGPnet Optionen aufrufen und im Karteireiter Authentification unter X.509 Authentification über den Button Select Certificate (siehe Abbildung oben) das importierte persönliche Zertifikat auswählen

---

Advanced



Allowed Remote Proposals

hier wird festgelegt, welche Verschlüsselungsalgorithmen, Hashalgorithmen, Datenkompression und Keylänge des Diffie-Hellmann Keys dem Kommunikationspartner zur Benutzung erlaubt sind.

- Ciphers: CAST oder Triple-DES Algorithmus zur Ver- und Entschlüsselung
  (siehe auch Kurzinfos zu verwendeten Algorithmen)
  None: es wird keine Verschlüsselung eingesetzt
- Hashes: SHA-1 oder MD5 Hash, der für den Authentifizierungsprozess eingesetzt wird
  (siehe auch Schlüssel)
  None: es wird keine Authentizitätsprüfung eingebunden
- Diffie-Hellmann: Keylänge des Diffie-Hellmann Keys, der für das *Authenticated Diffie-Hellman Key Agreement Protocol* (ein Abstimmungsverfahren, über das ein gemeinsamer Secret Key zweier Kommunikationspartner unter Hinzunahme von digitalen Signaturen und Public Key Zertifikaten gebildet wird) eingesetzt wird.
- Compression: LZS oder Deflate zur Kompression der Daten (sinnvoll bei Modem und ISDN Verbindungen, nicht sinnvoll bei Kabelmodem-, [A]DSL-, T-1- und T-3-Verbindungen)

Proposals

hier werden die eigenen Vorschläge, die man dem Kommunikationspartner zur Verschlüsselung und Authentifizierung der SA anbietet, festgelegt. Dazu kann man verschiedene Kombinationen aus zu verwendendem Verschlüsselungs- und Hashalgorithmus, Datenkompressionsart und Keylänge des Diffie-Hellmann Keys definieren.

Eine Kombination muss mit den Einstellungen, die der Kommunikationspartner in seinen *Allowed Remote Proposals* festgelegt hat, harmonieren.

IKE

- Authentication (Authentifizierung mittels): RSA oder DSS Signatur oder eines Shared Key (gemeinsamer Key)
- Hash: SHA-1 oder MD5
- Cipher (Verschlüsselung mit): CAST oder Triple-DES
- DH (Diffie-Hellmann Keylänge): 1024 oder 1536-bit

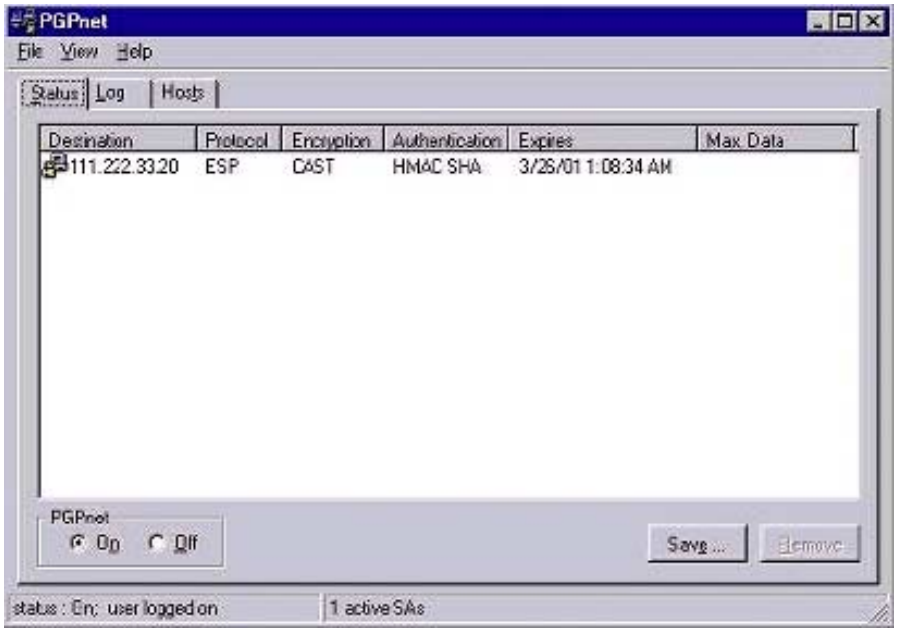
IPSEC

- AH (Authentication Header): IPSec-Unterprotokoll, dass nur die Authentifizierung verschiedener Teile eines IP Headers über SHA-1 oder MD5 Hashes regelt
- ESP (Encapsulating Security Payload): IPSec-Unterprotokoll, dass die Verschlüsselung per CAST oder Triple-DES Algorithmus und die Authentifizierung per SHA-1 oder MD5 Hash regelt
- IPPCP (IP Payload Compression Protocol): Datenkompresssion per LZS oder Deflate
- Perfect Forward Secrecy: Keylänge des Diffie-Hellmann Keys für alle definierten IPSec Proposals

Anwendung über das PGPnet Fenster

Aufruf über PGPnet Systemtray/PGPnet/Status, Log und Hosts

Status Fenster

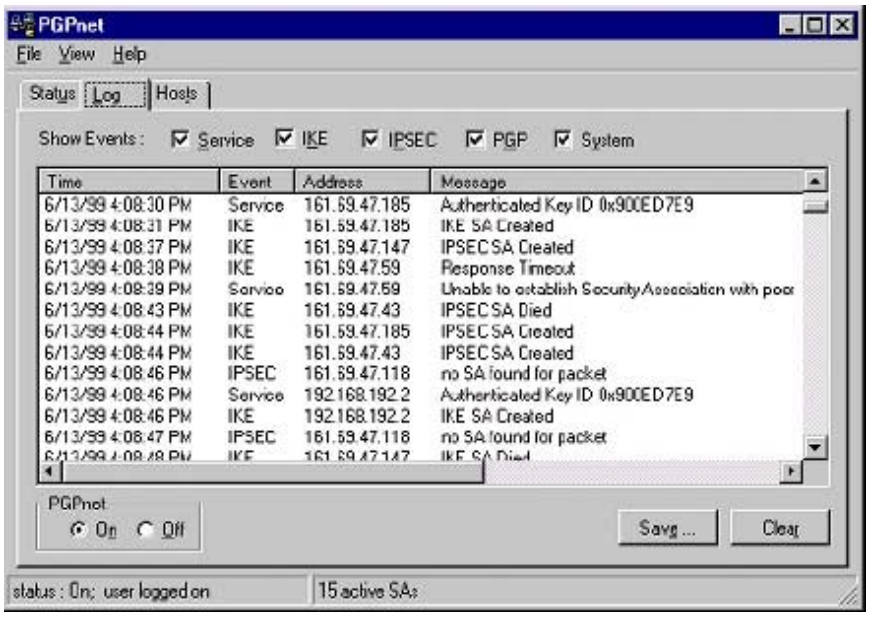


Im Status Fenster werden alle SA´s mit Angaben zur Gültigkeitsdauer, der Verschlüsselung und dem Ziel aufgelistet.
Das "Verhalten" einer SA wird durch die Optionen bestimmt, die man in den Advanced Optionen von PGPnet festlegt (siehe oben).

Folgende Angaben können aus den Spalten des Statusfensters gewonnen werden:

Destination (Ziel)	die IP Adresse des Zielrechners und/oder Gatewayrechners
Protocol (VPN Protokoll)	der Typ des miteinander ausgehandelten Protokolls: AH, ESP, IPCOMP
Encryption (Verschlüsselung)	der Typ des miteinander ausgehandeltem Verschlüsselungsalgorithmus: CAST oder Triple-DES. Wenn die SA nur der Authentifizierung dient, ist die Spalte leer
Authentication (Authentifizierung)	der Typ des miteinander ausgehandelten Authentifizierungsalgorithmus: HMAC MD5 oder SHA. Wenn im Proposal sowohl ESP als auch AH definiert sind, finden sich zwei Einträge
Expires (Ablauf)	Datum und Uhrzeit, wann die SA ungültig wird, "Never", wenn die Gültigkeitsdauer allein auf durch die Menge der übertragenden Daten basiert.
Max. Data	das Maximum an Daten in MB, das über die SA transportiert wird, bevor sie ihre Gültigkeit verliert.

Log Fenster

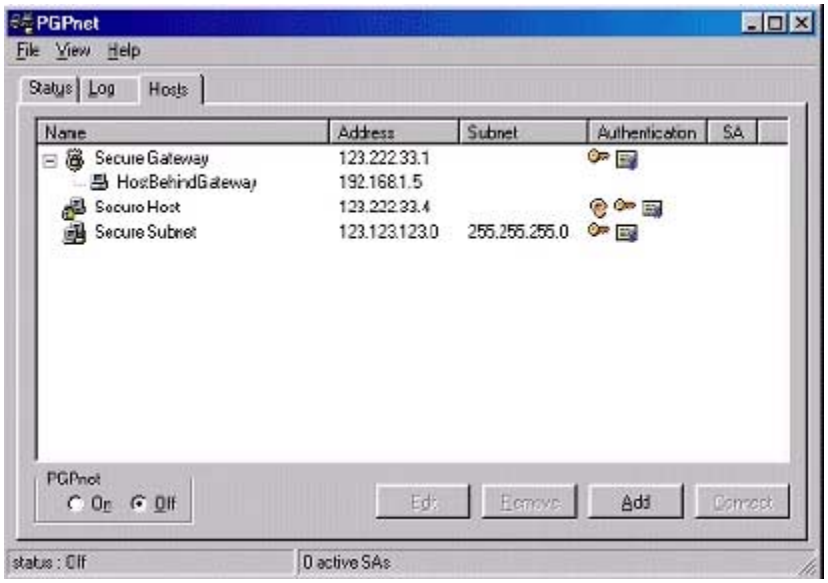


Das Log Fenster zeigt eine kurze Beschreibung aller Ereignisse und Fehler an. Über "Show Events" können Ereignisse die den Service, IPSec, PGP und/oder das System betreffen an- oder ausgeschaltet werden.
Über den Button "Save" kann der Inhalt des Logs als Textdatei abgespeichert und über "Clear" der Inhalt des aktuellen Logs gelöscht werden.

Time (Zeit)	Datum und Uhrzeit, wann das Ereignis/der Fehler eintrat
Event (Ereignis)	Typ des Ereignisses: Service, IKE, IPSec, PGP, oder System
Address (IP Adresse)	IP Adresse des Zielrechners
Message (Nachricht)	ein Text, der eine kurze Beschreibung des aufgetretenen Ereignisses oder Fehler liefert

---

Host Fenster



Das Host Fenster zeigt alle Hosts (Einzelrechner), Gateways (Zwischen-, Firewall-, Proxyrechner) und Subnetze (IP Adressenbereich eines Netzes, der mehrere Einzelrechner einschliesst), zu denen auf gesichertem Wege einer SA oder in jedem Fall eine ungesicherte Verbindung aufgenommen werden soll:

Name selbstgewählter Name des Subnetzes, Host- oder Gatewayrechners

Address IP Adresse des Hosts, Subnetzes oder Gateways

Subnet ist das Ziel ein Subnetz, wird hier die Netzmaske des Subnetzes angezeigt

Authentication der Typ der Authentifizierung als Icon

- ein Schlüssel symbolisiert Authentifizierung über Public Key Kryptografie
- ein Zertifikat symbolisiert Authentifizierung per X.509 Zertifikat.
- ein Ohr symbolisiert Authentifizierung über eine gemeinsame Passphrase (*shared secret*)
- kein Icon symbolisiert eine ungesicherte Verbindung

SA Anzeige eines grünen Buttons, wenn eine SA zum Zielrechner besteht

Erklärung der Buttons

Edit Aufruf des Konfigurationsmenüs, in dem der Zielrechner näher konfiguriert wird

Remove Entfernung des Ziel Eintrags

Add	Aktivierung des Konfigurationsassistenten zur Neuanlage eines Hosts, Subnetzes oder Gateways (im Expertenmodus wird sofort das Konfigurationsmenü gestartet)
Connect/Disconnect	über "Connect" wird eine SA aufgebaut, "Disconnect" beendet die SA, wenn zuvor der entsprechende Hosteintrag markiert wurde

---

das Konfiguration eines einzelnen Hosts (Zielrechner), eines Subnetzes oder Gateways

Hinweise

Zur Vereinfachung sollte der Expertenmodus abgeschaltet sein.
Wird PGPnet in einer Firmenumgebung eingesetzt, sollte die Host-Tabelle schon vom Netzwerkadministrator konfiguriert sein, da er über alle notwendigen Daten verfügt.
Wird zur Authentifizierung eine gemeinsame Passphrase (*Shared Secret*) genutzt, muss die gleiche Passphrase auf dem Zielrechner konfiguriert sein.
Je nachdem, wieviel Rechner und Subnetze einem TCP/IP Netz zugeordnet sind, werden die IP-Adressen in die Klassen A bis D unterteilt. Jeder Netzklasse ist eine Netzmaske zugeordnet. So bildet ein Klasse-C-Netz 254 Rechner ab und hat die Netzmaske 255.255.255.0, die Netzwerk-ID (das ist die erste Zahl einer IP-Adresse) liegt zwischen 128 und 191, ein Klasse-B-Netz bildet schon 65354 Rechner ab und hat die Netzmaske 255.255.0.0, die Netzwerk-ID liegt zwischen 192 und 223.
Ein Netz kann wiederum in mehrere Unter-, bzw. Subnetze aufgeteilt werden. In der Hostabbildung aus dem PGP Manual hätte das Klasse-C-Netz die IP-Adresse 123.123.0.0 mit der Netzmaske 255.255.255.0 , ein Subnetz trägt die IP-Adresse 123.123.123.0, ein Rechner in diesem Subnetz hätte z. B. die IP-Adresse 123.123.123.1
Mehrere verschiedene Netze können durch Gateway-Rechner miteinander verbunden werden, die die Aufgabe haben, die Informationen, bzw. die Datenpackete eines Netzes in ein anderes zu transportieren. Der Gateway kann auch mit Funktionen zur Filterung und Regelung der Zugangsbefugnisse für Daten aus anderen Netzen oder von anderen Rechnern ausgestattet sein, dann spricht man von einem Firewall-Rechner. Daraus folgt für die Anwendung und Benutzung von PGPnet dass man, je nachdem, ob alle Rechner eines Subnetzes, ein einzelner Rechner, oder ein Rechner hinter einem Gateway erreicht werden soll,

- die IP-Adresse oder den Namen des Hosts (Zielrechner)
- die IP-Adresse und die Subnetz Netzmaske des Ziel-Subnetzes
- die IP-Adresse oder den Namen des Gatewayrechners

bekannt sein muss

Eintrag eines Hosts oder Subnetzes

1. Button "Add"
2. Button "Weiter"
3. Checkbox "**Host**"
   oder
   Checkbox "**Subnet**"
   aktivieren
4. "Enforce secure communications" - wenn gesicherte Verbindungen (SA) verwendet werden
   "Allow insecure communications" - wenn ungesicherte Verbindungen verwendet werden
5. im Textfeld eine Bezeichnung für den **Host** (Zielrechner) *oder* das **Subnetz** eingeben
6. den Host Domain Namen oder die IP-Adresse des **Hosts** (Zielrechners)
   oder
   die IP-Adresse und die Netzmaske des **Subnetzes**
   eingeben
7. bei einer **ungesicherten** Verbindung:
   - Fertig stellen

   bei einer **gesicherten** Verbindung:

   b. "Use public-key cryptography only" - wenn die Authentifizierung nur über Public Keys durchgeführt werden soll
   c. Fertig stellen
   d. "First attempt shared secret security, then fall back to public-key cryptography" - wenn zuerst eine Authentifizierung über eine gemeinsame Passphrase versucht und danach erst auf Public Keys zurückgegangen werden soll
   e. Eingabe der gemeinsamen Passphrase in die Textfelder (die Passphrase wird im Klartext auf der Festplatte gespeichert)
   f. die Angaben festlegen, wie sich der eigene Rechner gegenüber dem Zielrechner identifiziert:
      IP address - mit der eigenen IP Adresse (000.000.000.000)
      Host Domain Name - mit dem Namen des Rechners (host.domain.netz)
      User Domain Name - mit der E-Mail Adresse (user@host.domain.netz)
      Distinguished Name - mit einem selbstdefinierten Textstrang, der sich an den Angaben eines Zertifikats anlehnt
      ("CN="user",_C=DE,_EMAIL=user@host.domain.netz")
   g. Fertig stellen

Eintrag eines Gateways

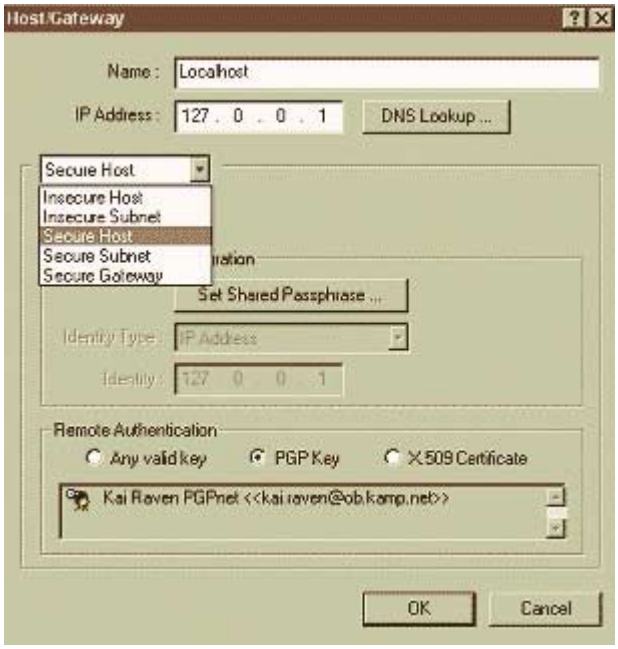
1. Button "Add"
2. Button "Weiter"
3. Checkbox "Gateway" aktivieren
4. "Enforce secure communications" - wenn gesicherte Verbindungen (SA) verwendet werden
   "Allow insecure communications" - wenn ungesicherte Verbindungen verwendet werden
5. den Host Domain Namen oder die IP-Adresse des **Gateways** eingeben
   a. "Use public-key cryptography only" - wenn die Authentifizierung nur über Public Keys durchgeführt werden soll
   b. Fertig stellen
   c. "First attempt shared secret security, then fall back to public-key cryptography" - wenn zuerst eine Authentifizierung über eine gemeinsame Pass-

phrase versucht und danach erst auf Public Keys zurückgegangen werden soll
   d. Eingabe der gemeinsamen Passphrase in die Textfelder (die Passphrase wird im Klartext auf der Festplatte gespeichert)
   e. die Angaben festlegen, wie sich der eigene Rechner gegenüber dem Zielrechner identifiziert:
      IP address - mit der eigenen IP Adresse (000.000.000.000)
      Host Domain Name - mit dem Namen des Rechners (host.domain.netz)
      User Domain Name - mit der E-Mail Adresse (user@host.domain.netz)
      Distinguished Name - mit einem selbstdefinierten Textstrang, der sich an den Angaben eines Zertifikats anlehnt
      ("CN="user",_C=DE,_EMAIL=user@host.domain.netz")
   f. die Abfrage, ob jetzt ein Host oder Subnet, die hinter dem Gateway liegen, mit dem Gatewayeintrag verbunden werden soll, bejahen oder verneinen.
      Bei der Bejahung der Abfrage startet der gleiche Dialog wie unter *Eintrag eines Hosts oder Subnetzes*,
      bei Verneinung der Abfrage wird der Gateway sofort eingetragen und Host oder Subnet können später (siehe unten) hinzugefügt werden.
   g. Fertig stellen

Eintrag eines Hosts oder Subnetzes, die sich hinter einem Gateway befinden

1. das Gateway ist schon konfiguriert im Hosts Fenster vorhanden
2. den Gatewayeintrag im Hosts Fenster markieren
3. Button "Add"
4. Button "Weiter"
5. Checkbox "Yes (*Create a new host entry which is behind the gateway XY*)" aktivieren
6. Checkbox "Host" oder "Subnet" aktivieren
7. "Enforce secure communications" - wenn gesicherte Verbindungen (SA) verwendet werden
   "Allow insecure communications" - wenn ungesicherte Verbindungen verwendet werden
8. im Textfeld eine Bezeichnung für den **Host** (Zielrechner) *oder* das **Subnetz** eingeben
9. den Host Domain Namen oder die IP-Adresse des **Hosts** (Zielrechners)
   oder
   die IP-Adresse und die Netzmaske des **Subnetzes**
   eingeben
10.
    a. "Use public-key cryptography only" - wenn die Authentifizierung nur über Public Keys durchgeführt werden soll
    b. Fertig stellen
    c. "First attempt shared secret security, then fall back to public-key cryptography" - wenn zuerst eine Authentifizierung über eine gemeinsame Passphrase versucht und danach erst auf Public Keys zurückgegangen werden soll
    d. Eingabe der gemeinsamen Passphrase in die Textfelder (die Passphrase wird im Klartext auf der Festplatte gespeichert)
    e. die Angaben festlegen, wie sich der eigene Rechner gegenüber dem Zielrechner identifiziert:
       IP address - mit der eigenen IP Adresse (000.000.000.000)
       Host Domain Name - mit dem Namen des Rechners (host.domain.netz)
       User Domain Name - mit der E-Mail Adresse (user@host.domain.netz)
       Distinguished Name - mit einem selbstdefinierten Textstrang, der sich an den Angaben eines Zertifikats anlehnt
       ("CN="user",_C=DE,_EMAIL=user@host.domain.netz")
    f. Fertig stellen

Wenn zu einem späteren Zeitpunkt die Angaben zu einem Host, Subnetz oder gatewayeintrag verändert werden müssen, kann man entweder den entsprechenden Eintrag im Hosts Fenster doppelt anklicken oder man klickt den "Edit" Button an.
Daraufhin öffnet sich das Editorfenster:



Hier kann ein DNS Lookup durchgeführt, der Status des Eintrags und die Authentifizierungsmethode abgeändert werden.
Im Abschnitt **Remote Authentication** kann für einen Eintrag zwingend vorgeschrieben werden, welchen PGP-Key oder welches X.509 Zertifikat der Rechner verwenden muss, um eine SA zum eigenen Rechner aufbauen zu können. Präsentiert der andere Rechner einen falschen Key oder ein falsches Zertifikat, wird eine SA dem Partner verweigert.
Wurde als Option die Erfordernis von validen Keys angegeben, muss der Key nach Überprüfung und Signierung als valid im Public Keyring gekennzeichnet sein, andernfalls kommt auch keine SA zustande.

Eine clevere Alternative zu PGPnet stellt das Programm BetweenUs dar, dass die Möglichkeit bietet, stark verschlüsselt, Textchats zu führen, Instant Messages abzusenden und Dateien zu transferieren.

PGP 5/6 in 8 Schritten für Eilige...

1. Nach der Installation widmen wir uns zuerst kurz den Grundeinstellungen von PGP

Optionen

Aufruf über Anklicken des PGPtrayicons, Menüpunkt "PGP Preferences"

GENERAL

Encryption and Signing Preferences (Verschlüsselungs- und Signieroptionen)

Always encrypt to default Key

Bedeutet, dass eine Datei, Text o.a. nicht nur mit dem öffentlichen Schlüssel (Public Key) des Empfängers, sondern zusätzlich noch mit dem eigenen Schlüssel verschlüsselt wird. So kann man selbst die verschlüsselte E-Mail später wieder entschlüsseln.
Aktivieren

Key Generation Preferences (Optionen zur Schlüsselerzeugung)

Faster Key generation

Betrifft die Erzeugung von DSS/Diffie-Hellmann Schlüssel, deren Länge fest vorgegeben ist. Bei der Errechnung des Schlüsselpaars bei diesen Längen wird ein im voraus berechneter Satz von Primzahlen benutzt, um die Geschwindigkeit zu erhöhen.
Deaktivieren

File Wiping Preferences
(Dateilöschoptionen)

Number of passes

Hier kann eingestellt werden, wie oft der Inhalt einer zu löschenden Datei überschrieben wird.
10 Durchgänge eintragen

FILES

In diesem Menü werden die Pfade zum öffentlichen und privaten Schlüsselring (Pubring und Secring) angegeben.
Man kann die Schlüsselringe auf der Festplatte oder auf einem Wechselmedium abspeichern.

EMAIL

Use PGP/MIME when sending email

Betrifft E-Mailprogramme, für die ein PGP 5.X Plug-In existiert und die die MIME Implementation von PGP unterstützen.
Aktivieren, wenn man selbst und der Empfänger so ein E-Mailprogramm benutzt

Word wrap clear-signed messages at column...

Damit wird eingestellt, ab welcher Breite der Zeilenumbruch der PGP Signatur bei E-Mails erfolgen soll, die unverschlüsselt versendet, aber mit einer PGP-Signatur versehen werden.
Generell sollte die Länge in PGP kürzer als im E-Mailprogramm eingestellt werden, z. B. 73 in PGP, 75 im E-Mailprogramm

Encrypt new mesages by default

Alle E-Mails werden immer an den Empfänger verschlüsselt, so lange ein öffentlicher Schlüssel des Empfängers vorhanden ist.
Für E-Mailprogramme, die den PGP/MIME Standrad und die PGP Plug-Ins unterstützen.

Sign new messages by default

Alle Postings und E-Mails werden immer signiert.
Für E-Mailprogramme, die den PGP/MIME Standrad und die PGP Plug-Ins unterstützen.

Automatically decrypt/verify when opening messages

Wird eine PGP verschlüsselte E-Mail geöffnet, wird sie, wenn gleichzeitig das Zwischenspeichern (Caching) des Passwortes eingestellt ist, automatisch entschlüsselt, bzw. die enthaltene Signatur geprüft.
Für E-Mailprogramme, die den PGP/MIME Standard und die PGP Plug-Ins unterstützen.

SERVER

Hier werden die Schlüsselverwaltungsrechner (Keyserver) eingetragen, die benutzen werden, um

- eigene, öffentliche Schlüssel zu verbreiten
- geänderte, öffentliche Schlüssel auf dem Keyserver abzugleichen
- Rückzugsurkunden kompromittierter öffentlicher Schlüssel zu veröffentlichen
- öffentliche Schlüssel anderer PGP Benutzer zu bekommen oder zu suchen
- eigene, öffentliche Schlüssel zu löschen (betrifft nur die Keys auf dem NAI eigenen Certification Server)

Alternativ kann man das WWW-Keyserverinterface zum Keymanagement benutzen

ADVANCED

Encryption

im Feld "Enabled algorithms" werden die Verschlüsselungsalgorithmen CAST, Triple-DES und IDEA aktiviert, die benutzt werden sollen, um Dateien oder Texte konventionell zu verschlüsseln oder bei der Erzeugung des eigenen öffentlichen Schlüssels und späteren Benutzung herangezogen werden können.
Im "Preferred algorithm" (zu bevorzugenden Algorithmus) IDEA wählen.

Trust Model

Display Marginal Validity Level

Über diese Option wird unter der Spalte "Validity" (Gültigkeit) der Grad der Authentizität, die man einem Schlüssel zubilligt, entweder in Form verschiedenfarbiger Knöpfe/Rauten oder verschiedenschraffierter Balken angezeigt.

Export Format

Compatible

Bei dem Export eines öffentlichen Schlüssels aus dem Pubring (und Speichern als Datei) wird ein Dateiformat verwendet, das zu den vorherigen PGP Versionen kompatibel ist. *Diese Option sollte immer aktiviert sein mit der Ausnahme, dass bekannt ist, dass der Empfänger ebenfalls PGP 6 benutzt.*

Complete

Bei dem Export eines öffentlichen Schlüssels aus dem Pubring (und speichern als Datei) wird das PGP 6 Dateiformat verwendet, das auch Angaben zu Foto-IDs und Designated Revokers enthält und nicht kompatibel zu vorherigen PGP Versionen ist.

2. Im nächsten Schritt erstellen wir einen RSA und einen Diffie-Hellmann/DSS Schlüssel

Um mit PGP verschlüsselte E-Mails austauschen zu können, brauche ich einen öffentlichen Schlüssel (Public Key) und einen privaten Schlüssel (Secret oder Private Key), beide zusammen bilden mein Schlüsselpaar. Den öffentlichen Schlüssel stelle ich allen Kommunikationspartnern zur Verfügung, damit sie mit meinem öffentlichen Schlüssel ihre E-Mails an mich verschlüsseln können. Den privaten Schlüssel behalte ich, denn damit entschlüssele ich wieder die mit meinem öffentlichen Schlüssel verschlüsselten E-Mails.

Vorgehensweise

1. im Kontextmenü des PGPtrayicons "PGPkeys" oder direkt im Startmenü auf das PGPkeyicon klicken
2. Menüpunkt "Keys"
3. "New Key"
4. Full Name: (Vorname) Nachname eingeben
5. E-Mail Address: die eigene E-Mailadresse in der Form *user@adresse* eingeben
6. Key-Typ RSA oder Diffie-Hellmann/DSS wählen -
7. Key Pair Size (Schlüssellänge) wählen
   bei Diffie-Hellmann/DSS: unter "Custom" 4096 bits eintragen
   bei RSA: 2048 bits wählen
8. unter "Key Expiration" (Schlüssel Ungültigkeit) "Key pair never expires" (Schlüssel ist immer gültig) wählen
9. jetzt in beiden Textfeldern ein Passwort eingeben, dass mindestens 20 Zeichen umfasst und aus keinen Wörtern besteht, die auch in einem Wörterbuch zu finden wären, z. B. in der Form wie *WedQu23_DaW///?Nurei*
10. Fertig stellen, dabei die automatische Versendung an den Keyserver erst einmal nicht benutzen

3. Im nächsten Schritt erstellen wir für die gerade erzeugten Schlüssel Rückzugsurkunden (Key Revocation)

Die Key Revocation dient dazu, den öffentlichen Schlüssel als zurückgezogen, bzw. ungültig zu erklären.
Warum jetzt schon ?
Es ist sinnvoll, bereits nach der Schlüsselerzeugung die Key Revocation für den eigenen öffentlichen RSA und Diffie-Helmman Schlüssel zu erzeugen, denn im Falle eines Verlustes oder Diebstahls des privaten Schlüssels und/oder des Passwortes, ohne vorher die Key Revocation erzeugt zu haben, kann der Schlüsselbesitzer die Key Revocation nicht mehr herstellen und damit seinen Schlüssel nicht als zurückgezogen, bzw. ungültig auf den Keyservern markieren. Eine Löschung des eigenen öffentlichen Schlüssel durch die Betreiber der Keyserver ist nicht möglich.

Vorgehensweise

1. beide Schlüsselbunddateien (pubring.*/secring.*) kopieren (z. B. Kopieren auf Diskette oder in ein anderes Verzeichnis)
2. PGPkeys aufrufen
3. den eigenen Schlüssel markieren
4. im Kontextmenü zum Schlüssel den Eintrag "Revoke" oder im Menü "Keys" den Eintrag "Revoke" anklicken
5. Sicherheitsabfrage mit "ja" bestätigen
6. Passwort eingeben, danach erscheint der eigene Schlüssel mit einem roten Querbalken im PGPkeys Fenster
7. den Schlüssel über Menü "Keys" Eintrag "Export" als Datei exportieren (z. B. als "RSArevoke.asc" und "DHrevoke.asc") - im Verzeichnisfenster keine weiteren Optionen aktivieren
8. die Dateien sicher vor unbefugtem Zugriff abspeichern
9. die im ersten Schritt gesicherten Schlüsselbundddateien komplett zurückkopieren und dabei die bestehenden Schlüsselbunddateien überschreiben

4. Im nächsten Schritt stellen wir unseren öffentlichen Schlüssel der Allgemeinheit zur Verfügung

Der eigene öffentliche Schlüssel muss veröffentlicht werden, damit Kommunikationspartner ihn zur Verschlüsselung von E-Mails benutzen können. Dazu speichern wir den öffentlichen Schlüssel zuerst als Datei ab und versenden die Schlüsseldatei an einen Keyserver (das ist ein Rechner, auf dem unzählige öffentliche Schlüssel der anderen PGP Benutzer gespeichert werden) oder per E-Mail an einen E-Mailpartner

Vorgehensweise

1. PGPkeys aufrufen
2. den eigenen Schlüssel markieren
3. im Menü "Keys" den Eintrag "Export" anklicken
4. Verzeichnis und Dateiname wählen
   die Checkboxen "Include Private Keys" und "Include 6.0 Extensions" **nicht aktivieren**
5. der öffentliche Schlüssel liegt als "MeinRSAPublicKey.asc" oder "MeinDSS/DHPublicKey.asc" vor
6. den kompletten Inhalt der Schlüsseldatei im WWW-Keyserverinterface zum Keymanagement einfügen und abschicken und/oder die Datei an eine E-Mail an den E-Mailpartner anhängen, bzw den Inhalt der Schlüsseldatei in eine E-Mail hineinkopieren und versenden

5. Im nächsten Schritt fügen wir den öffentlichen Schlüssel eines E-Mailpartners unserem Schlüsselbund hinzu

Den öffentlichen Schlüssel haben wir auf verschiedenen Wegen erhalten:

- ein E-Mailpartner hat uns seinen Schlüssel als E-Mail Anhang ("Dateiname.asc") oder Bestandteil einer E-Mail (in der E-Mail findet sich ein Schlüsselblock, der mit -----BEGIN PGP PUBLIC KEY BLOCK----- beginnt und mit -----END PGP PUBLIC KEY BLOCK----- aufhört) gesendet
- wir haben die Datei, die den Schlüssel enthält ("Dateiname.asc"), von seiner Homepage im WWW heruntergeladen
- der Schlüssel wird uns nach Eingabe einer Suchanfrage im WWW-Keyserverinterface zum Keymanagement im Browser angezeigt

Vorgehensweise

der Schlüsselblock wird angezeigt:

1. Kontextmenü von PGPtray aufrufen
2. Menüeintrag "(Use) current window"
3. Menüeintrag "Decrypt & Verify"
4. Button *Import*

der Schlüsselblock liegt als Datei vor:

1. im Explorer die Datei doppelt anklicken
2. Button *Import*

6. Im nächsten Schritt signieren wir eine E-Mail an einen Mailpartner

Mit der PGP-Signatur wird zum geschriebenen E-Mailtext mittels mathematischer Methoden eine Zeichenkette gebildet, die der Gesamtheit des Textes entspricht und somit eine Prüfsumme des Textes darstellt (Hash). Die Zeichenkette wird anschliessend mit dem eigenen, privaten Schlüssel verschlüsselt. Der Empfänger entschlüsselt mit meinem öffentlichen Schlüssel die Zeichenkette. Wurde der Text während der Übertragung verändert, entspricht der Wert der Zeichenkette nicht mehr dem Text, der Empfänger weiss nun, dass eine Veränderung, bzw. mögliche Fälschung vorliegt. Wurde der Text verändert und mit einem falschen Schlüssel signiert weitergeleitet, passt mein öffentlicher Schlüssel nicht mehr zur verschlüsselten Zeichenkette.

Vorgehensweise

1. Kontextmenü von PGPtray aufrufen
2. Menüeintrag "(Use) current window"
3. Menüeintrag "Sign"
4. Passwort für den eigenen Schlüssel eingeben

7. Im nächsten Schritt verschlüsseln wir eine E-Mail an einen Mailpartner

Vorgehensweise

1. Kontextmenü von PGPtray aufrufen
2. Menüeintrag "(Use) current window"

3. Menüeintrag "Encrypt"
4. im Schlüsselauswahlfenster (Recipient list) den Schlüssel des Empfängers mit der Maus in das untere Empfängerfenster (Recipients) ziehen
5. Button *OK*

Wenn man eine E-Mail sowohl signieren, als auch an den Empfänger verschlüsseln will, wählt man statt Sign oder Encrypt einfach den Menüeintrag "Encrypt & Sign" aus.

8. Im nächsten Schritt entschlüsseln wir eine E-Mail, die wir von unserem Mailpartner erhalten haben

Meistens werden wir E-Mails erhalten, deren Inhalt vollständig verschlüsselt ist, so eine E-Mail beginnt mit -----BEGIN PGP MESSAGE----- und endet mit -----END PGP MESSAGE-----
Nach der Entschlüsselung wird uns der Klartext von PGP angezeigt, ausserdem die User-ID (Name und E-Mailadresse) des Absenderschlüssels, falls unser Mailpartner die E-Mail zusätzlich signiert hat.
Wenn wir wiederum vorher den öffentlichen Schlüssel unseres Mailpartners auf Gültigkeit und Echtheit überprüft und ihn deshalb mit unserem privaten Schlüssel signiert hatten, wird zusätzlich der Status der Signatur als *good* oder *valid* ausgewiesen, andernfalls als *bad* oder *invalid*.

Vorgehensweise

1. Kontextmenü von PGPtray aufrufen
2. Menüeintrag "(Use) current window"
3. Menüeintrag "Decrypt & Verify"
4. Passwort für den eigenen Schlüssel eingeben

Probleme und Problemlösungen

Hier werden Probleme aufgeführt, die mit PGP 5/6 bei Anwendern auftraten und versucht, Möglichkeiten zu ihrer Lösung, falls möglich, aufzuzeigen.
D. h. nicht, dass die genannten Probleme immer und bei jedem Anwender auftreten, da Fehlermeldungen oder Probleme oftmals sehr abhängig von der Gesamtkonfiguration eines Systems sind (übrige Programme, Background-Tasks, Dienste usw.)
Da sich Probleme auch mit Programmen und Konfigurationen ergeben, die ich selbst nicht einsetze, würde ich mich über Zusendung weiterer Problemschilderungen **und** dazugehörigen Lösungswegen freuen, damit sie in die FAQ aufgenommen und so allen Benutzern zugänglich gemacht werden können.

Probleme

- Bei laufendem PGP 6.5.1 kann aus Word 97 nicht gedruckt werden, Word bricht mit einem GPF ab, nach Druck auf HP Printer erscheint GPF.
- PGPtray muss vor dem Drucken beendet werden.
- PGP kann nicht komplett deinstalliert werden, kann nicht neu installiert werden, weil sich Reste einer vorherigen Installation auf dem System befinden
- Um PGP manuell komplett zu deinstallieren, folgenden Weg einschlagen:
  Das PGP Verzeichnis komplett löschen, alle übrigen PGP Systemdateien aus dem \Windows und \Windows\System Verzeichnis löschen:

  \Windows\PGP_sdk.prf
  \Windows\PGPMacBinaryMappings.txt
  \Windows\PGPmemlock.vxd
  \Windows\PGPgroup.pgr
  \Windows\System\PGP60.hlp
  \Windows\System\PGP_SDK.dll
  \Windows\System\PGP60cl.dll
  \Windows\System\PGP60hk.dll
  \Windows\System\PGP60mn.dll
  \Windows\System\PGP60sc.dll
  \Windows\System\PGPsdkNL.dll
  \Windows\System\PGPsdkUI.dll
  \Windows\Applog\pgp*.*

  anschliessend alle Einträge zu PGP in der Registry (Start=>Ausführen=>Regedit) löschen, dabei als Suchbegriffe "PGP", "Network Associates", "McAfee" verwenden

- Mit PGP 6.0.2 und Outlook Express 4.0 funktioniert das PGP Plugin nicht, Outlook Express meldet, dass es *pgpmsmn.exe* nicht findet und zeigt die Fehlermeldung:
  The PGP Outlook Express plug-in was not installed correctly, or someone has deleted the Outlook Express program. Please re-install Outlook Express, then re-install the plug-in.
- Vor der PGP Installation die Datei *msimn.exe* sichern, dann PGP mit Plugin installieren, die Datei *msimn.exe* in *pgpmsmn.exe* umbenennen und in das Outlook Programmverzeichnis verschieben
- Ich habe PGP 6.0.2 installiert, aber es arbeitet nicht mit Outlook Express 5.0 zusammen, das ich mit dem Internet Explorer 5.0 mitinstalliert habe.
- PGP 6.0.2 arbeitet nur mit OE 4.0 zusammen, für OE 5.0 benötigt man PGP 6.5.X

- Arbeitet PGP auch mit Windows 2000 zusammen ?
- Ja, aber erst ab Version 6.5.2

Links und weitere Anleitungen oder
"Diese Anleitung war erst der Anfang"

Anleitungen

- NAI PGP 6.5 Manual in Deutsch (PDF)
- PGP-Anleitung für Anfänger
  Einfache und kurze Anleitung mit vielen Abbildungen. Gut geeignet für Einsteiger
- BeWare Utilities PGP
  Anleitung zum Gebrauch von PGP 5.0 unter BeOS
- Florian Helmberger's PGP omnium gatherum
  vollständige deutsche Übersetzung der PGP 2.6.3 Dokumentation plus Befehlreferenzkarte
- Ralf Kästner PGP
  Anleitung zum Gebrauch von PGP auf dem AMIGA
- Kryptographie und Datenschutz im Internet - Wie benutze ich PGP ?
  sehr anschauliche Seite, die mit vielen Screensshots die ersten Schritte mit PGP 2.6.3 und PGP 5.0 erläutert (Gerade für Eudorabenutzer zu empfehlen)
- Marvel's PGP-Fokus
  Anleitung für Anfänger, Infos für Profis
- Comp.security.pgp FAQ
  zu Pretty Good Privacy (>2.6.3) von Galactus in deutscher Übersetzung
- Yogi's PGP-Einsteigerseite
  PGP-Kurs zu PGP 2.6.3 von Jürgen Pötzsch
- Tom McCune's PGP Questions & Answers
- Tom McCune's PGP Q & A in deutscher Übersetzung v. B. Witte

Informationen

- Robert's Crypto & PGP Links
- P. Zimmermann's ehemaliges PGP.COM
- Die Internationale PGP Homepage
  von Stale Schumacher
- PGP Europa
- Network Associates PGP
- PGP-Public-Keyserver: UniGH Paderborn
- Raven's No Big Brother Web Page
- Deutschsprachige Übersetzung der RSA FAQ
  von Lutz Donnerhacke
- c't - Krypto-Kampagne - Linksammlung Pretty Good Privacy
- Pressemitteilung von Wirtschaftsminister Rexrodt vom 26.03.1998
  zu Kryptografiegesetzen
- An Open Specification for Pretty Good Privacy (OpenPGP)
  der IETF

Literatur

- Schmeh, Klaus: Safer Net. Kryptografie im Internet und Intranet.
  dpunkt Vlg. 1998. 456 S. ISBN 3-932588-23-1
- Wobst, Reinhard: Abenteuer Kryptologie. Methoden, Risken u. Nutzen d. Datenverschlüsselung.
  Addison-Wesley. 1997. Ca. 300 S. ISBN 3-8273-1193-4
- Smith, Richard E.: Internet-Kryptographie.
  Addison-Wesley. 1998. 384 S. ISBN 3-8273-1344-9
- Bauer, F. L.: Entzifferte Geheimnisse. Methoden u. Maximen d. Kryptologie.
  Springer. 1997. 472 S. ISBN 3-540-62632-8

- Beutelspacher, A. : Kryptologie.
  Vieweg Vlg. 1996. 178 S. ISBN 3-528-48990-1
- Schneier, Bruce: Angewandte Kryptographie. Protokolle, Algorithmen und Sourcecode in C.
  Addison-Wesley. 1997. 844 S. ISBN 3-89577-854-7
- Garfinkel, S.: PGP: Pretty Good Privacy. Verschlüsselung von E-Mail.
  O´Reilly. 437 S. ISBN 3-930673-30-4
- Weikert, A. : Kryptographie m. d. Computer. Verschlüsselungspraxis mit PGP.
  Pflaum. 1996. 93 S. ISBN 3-7905-1503-5
- Hagemann, H. [u.a.] : Kryptologie - Interaktives Training. Techn. Datenschutz i. Kommunikationsnetzen.
  Addison-Wesley. 1997. ISBN 3-8273-1220-5
- Creutzig, Christopher. Buhl, Andreas. FoeBuD e.V. [Hrsg.]: PGP - Pretty Good Privacy. Der Briefumschlag für Ihre elektronische Post - Deutschsprachige Anleitung zum internationalen Verschlüsselungsprogramm
  Art d'Ameublement Vlg. 4. völlig neu bearb. u. erw. Aufl. 320 S. ISBN 3-9802182-9-5

PGP-Versionen oder
"Welche Versionen gibt es, welche benötige ich ?"

Sind irgendwelche Angaben nicht korrekt oder bedürfen näheren Anmerkungen bitte ich um Rückmeldung

PGP Version [Quelle]	Implementationen					
	RSA Generierung	RSA Benutzung	DH Generierung	DH Benutzung	Features	Anmerkung
2.6.3 ia	ja/2048-bit	ja/2048-bit	nein	nein		
2.6.3 in	ja/8192-bit	ja/8192-bit	nein	nein	Unterscheidung der Keys nach Encryption (ENCR) und Signing (SIGN) möglich	wird von der Individual Network CA eingesetzt
5.0 Freeware	nein	ja	ja	ja	DH (ElGamal)/DSS Keys, PGPkeys eingeführt	CMRK-Keys werden eingeführt
5.0 Trialversion	ja	ja	ja	ja		auf 30 Tage begrenzt, Laufzeitpatch verfügbar
5.0 international	nein	ja	ja	ja		
5.5 Freeware	nein	nein	ja	ja	PGPtools eingeführt	CMRK-Keys weren in PGPkeys durch farbiges Icon hervorgehoben
5.5 Business Security	ja	ja	ja	ja		vollständig vorkonfigurierte PGP Klientversionen mit CMRK-Keys möglich
5.5.3 Freeware	nein	nein	ja	ja		RSA mit RSA Patch möglich
5.5.3 international	ja/2048-bit	ja/2048-bit	ja	ja		
5.5.3 iaLP	ja/8192-bit	ja/8192-bit	ja	ja	keine undendlichen Backups der Schlüsselringdateien	
5.5.3 Cyber Knight Templar (CKT)	ja/16384-bit	ja/16384-bit	ja/8192-bit/2048-bit DSS	ja/8192-bit/2048-bit DSS	erweitertes PGPlog	
5.5.3 Preston Wilson	ja	ja	ja	ja		

PGP Version [Quelle]	Implementationen					
	RSA Generierung	RSA Benutzung	DH Generierung	DH Benutzung	Features	Anmerkung
5.5.5 Personal Privacy	nein	ja	ja	ja		Version wurde wegen des Verkaufs von PGP Inc. an NAI eingeführt
5.5.5 Business Security (Email and Files)	ja	ja	ja	ja		
6.0 Business Security	ja/4096-bit	ja/4096-bit	ja	ja	PGPadmin Tool, PGPdisk 1.0 eingeführt	PGPdisk 1.0 besitzt Sicherheitsfehler
6.0 Personal Privay Desktop Security Freeware DH	nein	nein	ja	ja	Foto-ID, zusätzlicher Person als Revoker (designated revoker), bei DH unerschiedlicher Signier- und Verschlüsselungskey, Key Splitting, Use current window Funktion eingeführt	PGPdisk 1.0, für die Business Versionen wird die Bezeichnung Desktop Security eingeführt
6.0 Personal Privay Desktop Security RSA	ja/4096-bit	ja/4096-bit	ja	ja		PGPdisk 1.0
6.0 a Personal Privay RSA	ja/2048-bit	ja/2048-bit	ja	ja		PGPdisk 1.0
6.0 Personal Edition DH	nein	nein	ja	ja		PGPdisk 1.0
6.0.1 Personal Privacy Desktop Security RSA	ja/2048-bit	ja/2048-bit	ja	ja		PGPdisk 1.0
6.0.2 Personal Privacy	nein	ja/2048-bit	ja	ja	SecureViewer mit SoftTempest eingeführt	RSA nur mit 128-bit IE Securitypatch, PGPdisk 2.0

PGP Version [Quelle]	Implementationen					
	RSA Generierung	RSA Benutzung	DH Generierung	DH Benutzung	Features	Anmerkung
Desktop Security DH						
6.0.2 Personal Privacy Desktop Security RSA	ja/2048-bit	ja/2048-bit	ja	ja		PGPdisk 2.0, Outlook Express 4.0, Outlook 97/98 Plug-In
6.0.2 Personal Privacy RSA	ja/2048-bit	ja/2048-bit	ja	ja		PGPdisk 2.0, Outlook Express 4.0, Outlook 97/98 Plug-In
6.0.2 international	nein	ja	ja	ja		
6.0.2 CKT Build 5	ja/16384-bit	ja/16384-bit	ja/8192	ja/8192	zusätzlich RIPEMD-160,SHA-1,MD 5 als Signieralgorithmus frei wählbar, PGPdisk, erweitertes PGPlog, konfigurierbare Kommentarzeile	keine unendlichen Backups der Schlüsselringdateien, Bugfixes, PGPdisk 2.0, Outlook Express 4.0, Outlook 97/98 Plug-In
6.5.1 Freeware	ja/2048-bit	ja/2048-bit	ja	ja	eingeschränktes PGPnet, DH Kommandozeilenversion, biometrischer Wortlistenfingerprint, Hotkeys, selbstextrahierende, CAST verschlüsselte Archive (SDA) eingeführt	kein PGPdisk, nutzt RSAREF für RSA,Outlook 97/98/2000 und Outlook Express 4.0/5.0 Plug-In, nicht Windows 2000 fähig
6.5.1 international	ja/2048-bit	ja/2048-bit	ja	ja	eingeschränktes PGPnet, DH Kommandozeilenversion, biometrischer Wortlistenfingerprint, Hotkeys, selbstextrahierende, CAST verschlüsselte Archive (SDA) eingeführt	kein PGPdisk, Outlook 97/98/2000 und Outlook Express 4.0/5.0 Plug-In, nicht Windows 2000 fähig
6.5.1 Personal Privacy Eval	ja/2048-bit	ja/2048-bit	ja	ja	vollständiges PGPnet, unterstützt die Anforderung von X.509 Zertifikaten über NAI's Net Tools PKI, VeriSign's OnSite	PGPdisk 2.0 enthalten, nutzt BSAFE für RSA, Outlook 97/98/2000 und Outlook Express 4.0/5.0

PGP Version [Quelle]	Implementationen					
	RSA Generierung	RSA Benutzung	DH Generierung	DH Benutzung	Features	Anmerkung
					und Entrust CA's und deren Verwendung in VPN's	Plug-In, nicht Windows 2000 fähig
6.5.2 a Freeware	ja/2048-bit	ja/2048-bit	ja	ja	PGPnet arbeitet mit dem IPsec Client von Windows 2000 zusammen (Voraussetzung: High Encryption Pack), Unterstützung des Intel Random Number Generator (RNG) ab Pentium III, auomatische E-Mail Plug-In Installation, kein vollständiges PGPnet	Outlook 97/98/2000 und Outlook Express 4.0/5.0 Plug-In, läuft auf Windows 2000 Rechnern, aber ohne PGPnet
6.5.2 a Desktop Security	ja/2048-bit	ja/2048-bit	ja	ja	Plug-in für Lotus Notes 4.5x - 4.6x Clients, Notes Plug-in Server Wizard zur Konfiguration von Lotus Domino Server und der User Datenbanken, Plug-in für Novell GroupWise 5.2.3, 5.2.4 und 5.5.x	Outlook 97/98/2000 und Outlook Express 4.0/5.0 Plug-In, läuft auf Windows 2000, aber ohne PGPnet
6.5.3 Desktop Security	ja/2048-bit	ja/2048-bit	ja	ja	Hotkeys für alle Encrypt & Sign Operationen	Outlook 97/98/2000 und Outlook Express 4.0/5.0 Plug-In, läuft auf Windows 2000 Rechnern, aber ohne PGPnet
7.0 Personal Privacy	ja/4096-bit (?)	ja/4096-bit (?)	ja	ja	Personal Firewall, Packet-Filter mit 6 Schutzleveln, Personal IDS (Intrusion Detection System), IPPCP compression World-Renowned Verschlüsselung, Instant Messaging Verschlüsselung für ICQ, virtuelle, verschlüsselte Festplatten, RSA mit bis zu 4096-bit Keylänge, Twofish (256-bit), RIPEMD-160	Plug-Ins für MS Exchange, MS Outlook 97/98/2000, MS Outlook Express 4.x/5.x für Windows 95/98/NT/2000, Macintosh Lotus Notes 4.5.x/4.6.x/5.0, Qualcomm Eudora 4.x für Windows 95/98/NT/2000 und Macintosh Claris E-Mailer

PGP Version [Quelle]	Implementationen					
	RSA Generierung	RSA Benutzung	DH Generierung	DH Benutzung	Features	Anmerkung
GnuPG von W. Koch	(nein)	(nein)	ja	ja	komplementär zu RFC 2440 (OpenPGP), implementiert ElGamal (DH), DSA, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 und TIGER, neue Algorithmen können durch Erweiterungsmodule implementiert werden, plattformunabhängig	RSA und IDEA müssen aus patentrechtlichen Gründen selbst kompiliert werden, kein GUI - aber Windows GUI in der Entwicklung, nur eingeschränkte Windows Beta, teilweise komplizierte Kommandoeingaben nötig und Inkompatibilitäten zu PGP

PGP 5/6 und 2.6.3 Versionsübersicht

PGP 5.X Versionen

Die amerikanische Version PGP 5.X liegt als sich selbstextrahierendes und -installierendes ZIP Archiv in folgenden Versionen vor:

5.0 Freewareversion

- es ist nicht möglich, RSA-Keys zu erzeugen, bestehende RSA-Keys können aber importiert und verwaltet werden
- es gibt keine Zeitbeschränkung
- CMR-Public Keys werden ohne Hinweis oder Warnung benutzt.

5.0 Trial- oder Evaluationversion

- mit der Trialversion ist es möglich, RSA-Keys und DSS/Diffie-Hellmann Keys zu erzeugen, importieren und zu verwalten
- CMR-Public Keys werden ohne Hinweis oder Warnung benutzt
- Die Funktion der Verschlüsselung und Signierung wird nach 30 Tagen eingestellt, alle Funktionen werden nach 60 Tagen eingestellt, nach dem Ende der Testphase hilft entweder Kauf, Neuinstallation oder Patchen von PGP 5.0

5.0i Internationale Version

- komplette Übersetzung der amerikanischen Freewareversion, die einzigen Unterschiede liegen im Hinzufügen des Buchstabens "i", die Abänderung des Startbildes und der Standardkeyserver ist jetzt horowitz.surfnet.nl
- CMR-Public Keys werden ohne Hinweis oder Warnung benutzt

- Wichtiger ist das Voliegen des kompletten Sourcecodes zu dessen Analyse

5.5 For Business Security

- eine erweiterte Version, die zum Einsatz in "Firmen" gedacht ist.
- neben PGPtrays und PGPkey finden sich noch die Programme PGPtools, eine Schnellstartleiste für die Standardfunktionen von PGP und PGPadmin, mit dem das Verhalten und die Funktionen von PGP und seinen Public Keys bis ins Detail konfiguriert werden kann
- mit PGP 5.5 wird CMR/GAK für den Homeuser offensichtlich, für Firmenangestellte gefährlich
  CMR wird für eingehende E-mails nur für DSS/DH-Keys, für ausgehende E-mails für DSS/DH und RSA-Keys möglich.
- RSA-Keyerzeugung und konventionelle Dateiverschlüsselung ist möglich.

5.5.X Freeware

- mit 5.5.3 ist die Erstellung von RSA-Keys, wie in der PGP 5.0 Freewareversion, nicht mehr möglich, aber anders als in der 5.0 Freewareversion bedeutet die Version 5.5.3 die endgültige Abschaffung der Unterstützung des RSA-Algorithmus, da auch die Anwendung von importierten RSA-Keys nicht mehr möglich ist
- Abhilfe schafft ein Update für die PGP 5.5.3 Freewareversion, die die Funtionalität der RSA-Key Erzeugung und des Handlings von RSA-Keys wieder herstellt.

5.5.X Internationale Versionen

- die internationale Freewareversion PGP 5.5.3 i kann sowohl DH- als auch RSA-Keys bis zu einer Länge von 2048 Bits verwalten und erzeugen.
  Daneben gibt es auch noch die Version PGP 5.5.3 iaLP, die auf dem ZEDZ Server zu finden ist und RSA-Keys bis zu einer Länge von 8192 Bits verwalten und erzeugen kann.

5.5.3 a von Preston Wilson

- volle RSA Unterstützung, von den Keyringdateien wird nur ein Backup angelegt.

PGP 6.X Version

6.X Cyber-Knights Templar

- Erzeugung von RSA Keys bis 16384-Bits
- Erzeugung von DH Keys bis 8192-Bits
- zusätzliche Hashalgorithmen SHA-1 und RIPE-MD 160 zur Signierung (auch für RSA Keys, aber inkompatibel zu PGP 2.6.3)
- Handling von DSA/DSS Keys bis 2048-Bits
- Handling des SHA-Double Hash Algorithmus
- zusätzliche Angaben zu Key-ID und Keyänge bei der Keyauswahl
- erweitertes PGPLog Fenster mit Key-ID Anzeige
- Anzahl der Ringbackups konfigurierbar
- Angabe der Key-ID als zusätzlicher Kommentar möglich
- Angabe des Key Fingerprints im Kommentar möglich
- veränderbare Versionsangabe
- erweitertes Explorer Kontextmenü
- enthält Sam Simpson's "PGP DH vs. RSA FAQ"
- ab Version 6.0.2 PGPdisk 2.0 enthalten

6.0 Freeware

- Verwendung und Erzeugung von RSA-Keys nicht möglich
- zur User-ID kann ein Foto hinzugefügt werden
- der Key einer zweiten Person kann dazu ermächtigt werden, den eigenen Key zu revoken (designated revoker)
- ein DH Key kann in einen Signatur- und Verschlüsselungskey unterteilt werden
- Key Splitting, d. h. Aufteilung des Secret Keys in mehrere Teile wird unterstützt

PGP 6.0 for Business Security Version

- einzige 6er Version, welche die Verwendung und Erzeugung von RSA-Keys bis 4096 bit ermöglicht.
- PGP-Administrator Tool enthalten
- Festplattenverschlüsselungsprogramm PGPdisk 1.0, das mit CAST und SHA-1 arbeitet, enthalten

PGP 6.0 Personal Edition

- Festplattenverschlüsselungsprogramm PGPdisk 1.0, das mit CAST und SHA-1 arbeitet, enthalten

PGP 6.0.2 Desktop Security DH

- PGPdisk 2.0 und PGPadmintool enthalten, eine Verwendung und Erzeugung von RSA-Keys ist bis zu einer Länge von 2048 bit nur möglich, wenn gleichzeitig der Internet Explorer 4.X mit installiertem 128-bit Securitypatch installiert ist, da diese Version die Microsoft Crypto API verwendet.
  Mit Version 6.0.2 ist ein altes Feature der Version 2.6.3 wieder verfügbar: Die Verschlüsselung "for your eyes only", so dass ein entschlüsselter Text nur im Tool "Secure Viewer" betrachtet, aber nicht im Klartext gespeichert werden kann. Bei der Anzeige wird ein SoftTempest Font, einer Entwicklung von Ross Anderson und Markus Kuhn von der Universität Cambridge, eingesetzt, der TEMPEST-Attacken erschweren soll.

PGP 6.0.2 Personal Privacy RSA

- PGPdisk 2.0 enthalten, diese Version erlaubt die Verwendung und Erzeugung von RSA-Keys bis 2048 bit.

PGP 6.0.2 Desktop Security RSA

- PGPdisk 2.0 und PGPadmintool enthalten, Erzeugung und Verwendung von RSA-Keys bis 2048 bit möglich.

PGP 6.0.2 i Internationale Version

- die Erzeugung von RSA-Keys ist nicht möglich

PGP 6.5.1 Personal Privacy/Desktop Security

- Erzeugung und Benutzung von RSA-Keys bis 2048-bit
- Selbstextrahierende, konventionell mit CAST verschlüsselte Archive

- VPN PGPnet mit verschlüsselten und/oder getunnelten TCP/IP Verbindungen über das PGPnet Protokoll, das auf IPSec (Internet Protocol Security) und IKE (Internet Key Exchange) Protokoll basiert.
- X.509 Authentifizierung, Benutzung von Secure Gateways und Tunneln von Verbindungen (nicht möglich mit PGP 6.5.1 Freeware)
- Automatisches Löschen freier Festplattenbereiche über Integration in den Windows Taskplaner
- Hot Keys für alle Verschlüsselungs-, Entschlüsselungs- und Signierfunktionen über die "Use current window" Option
- Darstellung des PGP-Fingerprints als Wortreihe
- verbesserter Zeilenumbruch (in Zitaten)
- HTTP Keyserver können über Proxies angesprochen werden
- PGPdisk (nicht in PGP 6.5.1 Freeware enthalten)
- DOS Version von PGP 6.5.1 ("Command Line Version") enthalten (nur DH/DSS)
- laut WinUsersGuide Unterstützung von Outlook 97/98 und Outlook Express 4.0, laut WhatsNew Readme auch Unterstützung von Outlook 2000 und Outlook Express 5.0

Hinweis:
Die Version von PGPdisk 1.0 besitzt nach Aussage von NAI ein Sicherheitsleck, dass erst ab PGP version 6.0.2 behoben wurde. Aus diesem Grund ist vom Gebrauch von PGPdisk 1.0 abzuraten.

PGP 2.6.3 xy Versionen

2.6.3 ia
die nichtmodifizierte, normale Version, kann RSA-Keys bis zu einer Länge von 2048 Bits erzeugen

2.6.3 (8192-bit)
modifizierte 2.6.3 ia Version, die RSA-Keys in den Längen 768, 1024, 2048, 4096 und 8192 Bits erzeugen kann

2.6.3 in
2.6.3 ia Version, die von der Individual Network Certification Authority (IN-CA) benutzt wird.
Auch von 2.6.3 in gibt es von Frank Prüfer eine Version, die Keys bis zu 8192 Bit erstellen kann
Nähere Information siehe Zertifizierungsstellen

E-mail und Newsreaderprogramme mit integrierter PGP 5/6 Unterstützung

- Calypso (bis PGP 6.5.3)
- The Bat
- Eudora
- AK-Mail ab V. 3.1
- Star Office 5.1 a
- Outlook (Express)
  Zusatzinformationen:
  Outlook Express 5.0 wird erst ab PGP Version 6.5.1 mit Plug-Ins unterstützt.

Die internationalen Versionen und der PGP Fond von Stale Schumacher
Da ein legaler Export starker Kryptografieprogramme in binärer Form aus den USA aufgrund der amerikanischen Exportkontrollgesetze nicht erlaubt ist, wird der Programmquellcode jeder PGP Freeware Version in Buchform ausgeführt, was erlaubt ist.

Alle Seiten der Codebücher werden dann in Norwegen von einem Team um Stale Schumacher eingescannt und nach Korrekturlesevorgängen zur internationalen Version "PGP X.Y.Z i" neukompiliert. Abgesehen von diesem Vorgang gibt es keinen Unterschied zu den angebotenen Funktionen und Verschlüsselungsstärken, bzw. -algorithmen der amerikanischen Originalversion PGP Freeware. Zur Unterstützung der Arbeit zum Kauf, Scannen und Bearbeiten der PGP Sources und benötigter Hard- und Software, um die internationalen PGP Versionen veröffentlichen zu können, hat Stale Schumacher einen PGP Unterstützungsfond eingerichtet.

Wer Geldbeträge in den Fond einzahlen will, kann Stale eine E-mail an den Exportfond schreiben, welche folgende Informationen enthält:

- VISA oder MasterCard Nummer
- Name des Kartenbesitzers
- Ablaufsdatum
- Höhe des Betrags

Die E-Mail sollte mit einer der Public Keys (0xCCEF447D (RSA) or 0x0A791610 (DSS/DH)) von Stale verschlüsselt werden.
Der Zahlungseingang wird von der Firma "Hypnotech AS", bei der Stale in Norwegen beschäftigt ist, gegengezeichnet. Überschussbeträge leitet Stale an die Electronic Frontier Foundation, Computer Professionals for Social Responsibility oder ähnliche Organisationen weiter (Vorschläge an Stale willkommen).