Inhaltsverzeichnis

Vorwort der Herausgeber

Von der E-Card zur E-Mail

Vergehen und Verbrechen gegen die „Privatheit“ sind heimtückisch: Die Auswirkungen sind zwar zu spüren, aber selten im Zusammenhang mit der Ursache. Die technische Entwicklung und ihre Nutzung ist dissipativ – sie läßt sich nicht umkehren. Alles was wir heute tun und zulassen, werden wir auch in Zukunft ertragen müssen. Nahezu jeder Vorgang in unserem Leben, was wir einkaufen, wann wir an welchem Automaten Geld abheben, wann wir die Toilettenspülung drücken, wann wir uns verlieben, ist fernabfragbar. Die Generation der Großväter war der ethischen Überzeugung, daß diese Daten, die ja meist nur als Abfallprodukt der Abrechnung anfielen, nicht zur passiven Spionage im Alltag genutzt werden würden. Die Enkel jedoch sehen gerade die Auswertung und den Verkauf dieser Daten als Schlüsseltechnologie um „richtig dick Schotter zu machen“. Schon bald werden Sie im gehobenen Dienst keinen Job mehr bekommen, ohne daß der designierte Arbeitgeber vorher eine Profilaufstellung von Ihnen in Auftrag gibt. Der Menschheitstraum vom freien Leben innerhalb einer starken Gemeinschaft rückt damit noch ein Stückchen weiter in die Ferne. Nur die Spießer werden – egal ob mit Tattoo und Piercing oder ohne – weiterhin ein kleines, klägliches Leben führen können. Bis der Terror auch sie erreicht.

Die Entwicklung ist schleichend – aber rasant. Mit Zugriff auf eine einzige Datenbasis, nämlich z. B. die der Firma EDS (Electronic Data Systems) könnten Sie nahezu jeden ‚zivilisierten‘ Menschen auf dieser Welt ‚in den Griff kriegen‘. Sie wüßten, wann er gereist ist, mit wem, neben wem er im Flugzeug gesessen hat, wie er sein Ticket bezahlt hat, wann er Geld abgehoben hat, wer vor ihm oder nach ihm am selben Geldautomaten Geld abgehoben hat und wüßten, was er im Restaurant bestellt. Sie kennen seine Lieblingsfarbe und seinen Kreditrahmen. Stellen Sie sich vor: Auch andere Menschen könnten plötzlich mehr über Sie wissen, als Sie selbst.

„Mir doch egal?“ – Oder denken Sie einfach nur, daß Ihnen gerade kein Argument gegen diese Entwicklung einfällt? Sie brauchen nicht zu

begründen, warum Sie niemand Unbefugtes Ihre Briefe lesen lassen. Sie brauchen nicht zu erklären, warum Sie anderen verbieten, Daten über Sie zu speichern und zu verarbeiten. Es ist Ihr Leben und Sie und Ihre Privatheit sind kein Eigentum anderer. Sie sollten sich nicht an das ‚komische Gefühl in der Magengrube' gewöhnen.

Sie wollen eigentlich so berühmt sein wie Marilyn Monroe? Und noch kennt Sie keiner? Nein, dadurch, daß Sie Ihre Privatsphäre aufgeben werden Sie nicht berühmt. Und sicher kennen Sie auch das alte Sprichwort: Reden ist Silber – Schweigen ist Gold. Es nützt Ihrem eigenen Leben mehr, wenn Sie nichts fremdbestimmt von sich preisgeben. Im Gegenteil: Es nützt Ihnen letztendlich mehr, wenn Sie auch die Daten anderer schützen.

„Gebt Ihnen doch die Daten – das kann doch sowieso niemand auswerten." Falsch. „Data Mining" ist einfach. Sie haben es vielleicht selbst schon gemacht, wenn Sie eine Internetsuchmaschine nutzen. Sie geben drei Begriffe ein mit ein paar Pluszeichen davor und statt einigen Millionen sehen Sie plötzlich nur noch drei Treffer. Das ganze in wenigen (Milli-)Sekunden.

„Ich kann ja doch nichts tun." – Falsch. „Jede und jeder hat einen Einfluß größer als null." Unterschätzen Sie das nicht. Ihr Denken, Bedenken und besonders Ihr Handeln beeinflußt die Umgebung weit mehr, als sich die meisten Menschen vorstellen. Eine Umgebung von Duckmäusern wird Duckmäuser erzeugen. Eine geistig anregende, freie Gesellschaft wird geistig anregende, freie Menschen bedingen. Und die Ausnahmen bestätigen dann lediglich die Regel.

Lassen Sie sich nicht erzählen, daß Sicherheit, Ordnung und unsere Demokratie durch Verschlüsselung gefährdet seien: „Es hat keinen Sinn, die Demokratie dadurch zu schützen, indem wir sie abschaffen." Dieser Satz von INGO RUHMANN ist schlicht, einfach und richtig. Und wenn der Schutz der Privatsphäre kriminell wird, haben nur noch Kriminelle Privatsphäre. Ab wann aber zum Beispiel ‚Global Management' kriminell ist, müssen Sie als ‚Souverän' bestimmen. Denn gar nicht mehr so sehr ‚der Staat' ist der ‚Big Brother', sondern wir brauchen endlich wieder mehr Staat (eine wehrhafte Demokratie), die verhindert, daß wir zulassen, daß der letzte Rest Kontrolle über die Kapital- und Machtinteressen vollends aus den Händen der Gesellschaft genommen wird. Liberal ist ok. Aber der real existierende Liberalismus ist mindestens genauso gefährlich wie die Nachbarn Kapitalismus, Nationalismus und Kommunismus.

Gedankenstrich ...

Dies ist nun schon die vierte völlig neu bearbeitet Version dieses Buches. Wir sind glücklich über und dankbar für den riesigen Arbeitseinsatz, den CHRISTOPHER CREUTZIG geleistet hat. Er hat neben der inhaltlichen Arbeit auch noch das Layout (mit TEX) gemacht, die CD zusammengestellt und ebenso wie ANDREAS BUHL, der den Windows-Teil schrieb, hat er unglaublich viel ehrenamtliche Lebensenergie in dieses Buch gesteckt. Also lesen Sie das jetzt aber auch bitte gründlich und stecken Sie Ihre Datenpostkarten zukünftig in Briefumschläge. Denn ohne PGP versandten Sie bisher keine E-Mails sondern lediglich E-Cards. Allerdings können wir Ihnen nicht alle Mühe abnehmen. Auch Sie müssen sich ein wenig plagen, PGP und die Verschlüsselung mit Öffentlichem und Privatem Schlüsseln zu verstehen. Es gibt keine Sicherheit mit „Klick&Go“. Falsch verstandene und fahrlässig verwendete Verschlüsselung kann den Unsicherheitsgrad noch um den Faktor erhöhen, daß Sie sich nun sicher glauben, aber dennoch ausspioniert werden. In einer Firma, deren Patentunterlagen beim Versand per E-Mail in die Hände der Konkurrenz fallen, kann das unangenehm sein ...

Kein Meister wird vom Himmel fallen. Und auch wenn Sie der kluge Kopf in Ihrer Abteilung sind, so müssen es ja auch alle anderen Mitarbeiterinnen und Mitarbeiter verstehen. Falls Sie weiter Hilfe benötigen, so möchte ich Sie hier auf die Struktur von Seminaren aufmerksam machen, die Ihnen den Einstieg und das Nutzen von PGP erleichtern. Sie finden Informationen dazu unter `www.foebud.org/seminare`. Bei Drucklegung werden solche Seminare von unseren Partner in Nürnberg, Leipzig, München und von uns in Bielefeld angeboten.

Willkommen im Netzwerk des Vertrauens

Rena Tangens & padeluun

Vorwort

Im Informationszeitalter, das nach allgemeiner Auffassung immer noch gerade beginnt, sind Daten und Informationen das wirtschaftlich wichtigste Gut. Um mit ihnen arbeiten zu können, sind sowohl die Authentizität und Integrität der Daten, also die gesicherte Erkenntnis, daß die vorliegenden Daten in genau dieser Form tatsächlich vom angegebenen Absender stammen, als auch die Geheimhaltung vertraulicher Daten unabdingbare technische Voraussetzungen. Gerade für Daten, die über das Internet verbreitet werden, ist beides normalerweise nicht gesichert.

PGP schließt diese Lücke. Es verwendet moderne, gute Verfahren für digitale Unterschriften und für die Verschlüsselung vertraulicher Daten, auch ohne daß Sender und Empfänger irgendwelche weiteren Absprachen treffen müßten.

Die Tatsache, daß PGP bestens für den Einsatz durch Jedermann geeignet ist und im Gegensatz zu anderen Lösungen keinerlei zentrale Instanz voraussetzt, mit deren Glaubwürdigkeit das gesamte System steht und fällt, macht es zu einem wichtigen demokratischen Werkzeug, denn PGP gestattet dem mündigen Bürger, sein Grundrecht auf informationelle Selbstbestimmung selbst in die Hand zu nehmen, also selbst zu entscheiden, wer welche Informationen von ihm oder ihr verlangt. Da dies nicht von allen Entscheidungsträgern in Politik und Wirtschaft als unbedingt wünschenswert angesehen wird, sind Konflikte mit Regierungen gewissermaßen vorprogrammiert. Glücklicherweise ist die deutsche Bundesregierung kein solcher Konfliktgegner; in den USA sieht die Lage sehr viel weniger rosig aus.

Diese Freiheit hat natürlich ihren Preis: Wie bei jedem Werkzeug, das dem einzelnen Endanwender viele Möglichkeiten bietet, lassen sich auch bei PGP durch Unkenntnis gravierende Fehler begehen. Im Gegensatz zu Tresoren, Bandschleifern, Briefumschlägen und Bildbearbeitungen ist es bei einem Verschlüsselungssystem aber für den Laien schwierig, die eigenen Fehler zu entdecken, bevor es zu spät ist. Daher möchte dieses Handbuch das nötige Wissen vermitteln, um PGP erfolgreich einzusetzen.

Dieser Text ist ursprünglich als Übersetzung des US-englischsprachigen Handbuchs zu PGP 2.3a entstanden. In der Zeit seit Erscheinen jener Version ist viel geschehen, PGP hat sich weiterentwickelt und die politische Lage hat sich drastisch verändert; nicht zuletzt ist PGP ein kommerzielles Produkt geworden. Um diesen Änderungen Rechnung zu tragen, haben wir die vierte Auflage, die Sie in Händen halten, komplett überarbeitet und in großen Teilen neu geschrieben. Dabei haben wir uns vom ursprünglichen Text so weit gelöst, daß es nicht mehr angemessen ist, PHILIP ZIMMERMANNs Aussagen, wie in den ersten drei Auflagen, in der ersten Person stehenzulassen.

Paderborn, im Oktober 1999
Christopher Creutzig
ccr@foebud.org

Teil I.

Allgemeines

1. Über diesen Text

1.1. Leseanleitung

Dieses Buch beschreibt eine Reihe ähnlicher Programme. Zum Einen sind da verschiedene Versionen von PGP zu nennen, im Wesentlichen die Version 2.6.2i für nahezu alle Betriebssysteme (MS-DOS, Unix, Linux, VMS, MacOS, Amiga, Archimedes, Atari, ...), des Weiteren die Versionen 5.0, 5.5.3i und 6.0 für MS Windows. Darüber hinaus behandelt dieses Buch auch eine „Open Source"-Implementation des OpenPGP-Standards, ein Programm namens GnuPG (für Unix, Linux, MS Windows). Im allgemeinen Teil und auch in Teil II finden Sie Formulierungen der Art „PGP arbeitet so: ...". Diese allgemeinen Beschreibungen schließen – sofern nicht aus dem Kontext anders deutlich – auch GnuPG mit ein.

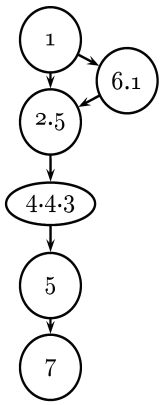


Abbildung 1.1: Minimal-Lese-Reihenfolge für Teil I

Welche Kapitel dieses Buches Sie in welcher Reihenfolge lesen sollten, hängt außer von Ihren Vorkenntnissen auch davon ab, welche Version von PGP Sie auf welchem Betriebssystem einsetzen möchten. Allen Lesenden wird dringend angeraten, dieses einleitende Kapitel komplett zu lesen, um im weiteren Text zu verstehen, was mit bestimmten Begriffen gemeint ist. Wenn Sie sich fragen, was und wozu Sie verschlüsseln sollten, empfehlen wir Abschnitt 6.1. Weiterhin sei allen Menschen, die sich noch nicht eingehend mit Verschlüsselungssystemen beschäftigt haben, Abschnitt 2.5 ans Herz gelegt, um einen Überblick zu erhalten, was PGP eigentlich tut. Bedauerlicherweise ist es nicht möglich,♠ echte Sicherheit zu erlangen, indem man irgendwo ein paar Knöpfchen drückt, ohne zu verstehen, was passiert. Von Kapitel 4 ist beim ersten schnellen Überfliegen zumindest Abschnitt 4.4.3 ratsam. Kapitel 5 ist kaum zu

♠ Das liegt nicht an PGP. Es ist prinzipiell nicht möglich.

vermeiden, wenn die durch PGP erreichte Sicherheit nicht ausgesprochen trügerisch sein soll. Kapitel 7 gibt Ihnen das notwendige Wissen an die Hand, um mit Personen, die Sie nie getroffen haben, eine gesicherte Kommunikation aufzubauen. Außerdem bewahrt die Lektüre jenes Kapitels Sie hoffentlich vor einigen schwerwiegenden „Anfängerfehlern“.

Erst nach diesen Hintergrundinformationen ist es sinnvoll, sich die Installation und Bedienung des Programms anzuschauen. Falls Sie sich noch nicht sicher sind, welche Version Sie einsetzen sollten, gibt hoffentlich Anhang A die nötigen Informationen für die Entscheidung. Für diejenigen unter Ihnen, die eine kommandozeilenorientierte PGP-Version (und evtl. anschließend ein graphisches Frontend) installieren wollen, ist der Teil II geschrieben. Diejenigen, die lieber eine „reine“ Windows-Applikation haben möchten, finden die nötigen Informationen in Teil III. Beide Teile setzen die Kenntnis des allgemeinen Teils voraus.

Informationen zur Lizensierung von PGP, Distribution, Copyrights, Warenzeichen, Gewährleistungen und Exportbeschränkungen enthält der Anhang E auf Seite 282.

Noch ein paar Worte über die Verwendung maskuliner und femininer Formen bei Begriffen wie „Anwender“ oder „Programmiererin“: In dieser Anleitung werden maskuline und feminine Formen austauschbar verwendet. In jedem Fall sind beide Geschlechter gemeint. Nebenbei bemerkt versuchen wir, nach alter Rechtschreibung zu schreiben, Abweichungen davon sind also echte Fehler und nicht einem Versuch der Umstellung zuzuschreiben. Das soll keine Rechthaberei sein – wir sind nur in der neuen Rechtschreibung bei weitem nicht sattelfest genug, um Ihnen diesen Text in der Qualität zu geben, die wir selbst von uns erwarten.

1.2. Übersetzung und Fachbegriffe

Bei Übersetzungen und deutschen Texten zu EDV-Themen ist es häufig schwer, zu entscheiden, in welchem Umfang Fachbegriffe eingedeutscht werden sollten. Werden zu viele Begriffe übersetzt, kann das Leserinnen verwirren, denen die englischen Begriffe vertraut sind. Umgekehrt kann ein zu wenig eingedeutschter Text für diejenigen unverständlich bleiben, die die englischen Begriffe nicht kennen. Wir haben uns für eine recht weitgehende Eindeutschung entschieden. Um nicht zuviel Verwirrung zu stiften, finden Sie in Tabelle 1.1 eine Liste der Worte, die Ihnen evtl. in

der englischen Form geläufiger sind, die wir aber in der deutschen Form verwenden.

englischer Begriff	unsere Übersetzung
default	Standard
directory	Verzeichnis
environment variable	Umgebungsvariable
exit code	Beendigungscode
file name extension	Nameserweiterung/Suffix
fingerprint	Fingerabdruck
grassroot organization	politische Basisorganisation
key compromise/revocation certificate	Schlüssel-Rückrufurkunde
key ring	Schlüsselbund
message digest	Textprüfsumme
pass phrase	Mantra
public key ring	öffentlicher Schlüsselbund
private key ring	geheimer Schlüsselbund
swapfile	Auslagerungsdatei

Tabelle 1.1: Von uns verwendete Übersetzungen

Weitere Fachausdrücke finden Sie im Glossar ab Seite 292 erläutert. Wir haben uns Mühe gegeben, Begriffe mit wohldefinierter Bedeutung zu wählen. Formulierungen wie „10 verlorene Bereiche in 8 Bereichen“ (dies verlautbarten ältere Versionen des MS-DOS-Programms `chkdsk`) haben sich hoffentlich nicht eingeschlichen.

1.3. Konventionen

In diesem Buch verwenden wir die folgenden Darstellungen: *Wichtiges* wird kursiv geschrieben, `Eingaben` und `Ausgaben` werden, ebenso wie `Menüeinträge`, in Schreibmaschinenschrift dargestellt. Angaben der Form [Sch95] sind Literaturangaben, das Literaturverzeichnis finden Sie im Anhang ab Seite 290. In der Kopfzeile finden Sie stets die Angabe des aktuellen Kapitels und Abschnittes darin einschließlich der Nummer des Buchteils, in dem Sie gerade lesen.

1.4. Danksagungen

An dieser Stelle möchte ich[△] zunächst PHILIP ZIMMERMANN danken, daß er PGP geschrieben und der Allgemeinheit zur Verfügung gestellt hat. Die Existenz dieses Programmes hat vieles überhaupt erst ermöglicht, nicht zuletzt dieses Buch. Ebenso gebührt den weiteren Entwicklern PGPs mein Dank. MARC AUREL möchte ich danken, daß er mich vor vielen Jahren auf meinen ersten Schritten mit PGP begleitet hat. Die erste Übersetzung der Anleitung zu PGP 2.3a (der Ur-Ur-Großvater des Buchs, das Sie in Händen halten) begann als Gemeinschaftsprojekt von sechs Leuten, von denen schließlich ABEL DEURING und ich übrigblieben.

Für die Verbreitung PGPs in Deutschland waren mehrere Leute wesentlich. Zunächst müssen wir uns hier selbst nennen, den FoeBuD e. V., der nicht nur diese Anleitung verbreitet hat (als PGP noch kaum bekannt war), sondern darüber hinaus sowohl durch Schulungen als auch politische Arbeit und nicht zuletzt ausführliche Betreuung von Journalisten und Politikern an der Schaffung des nötigen Rahmens mitgewirkt hat. Außerdem sind hier von den vielen beteiligten Menschen LUTZ DONNERHACKE für seine Arbeit mit dem Aufbau einer Zertifizierungsstruktur und PGP 2.6.3in und KAI RAVEN für seine Einsteiger-Anleitung[⊖] zu erwähnen.

Für die aktuelle Auflage geht mein Dank an DONALD E. KNUTH für sein System TEX und an LESLIE LAMPORT und das LATEX3-Team für LATEX 2ε; außerdem MARKUS KOHM für KOMA-Script. Typographische Fehler sind sicherlich meinen Änderungen zuzuschreiben.

Den Korrekturleserinnen und Korrekturlesern möchte ich hier besonders danken, ohne sie wäre dieses Buch zwar eher fertig geworden, aber inhaltlich gewonnen hat es auf jeden Fall. Besonders zu nennen sind hier PADELUUN, TOM BUDEWIG, URSULA BEGEROW und JÖRG JENETZKY. Weitere wichtige Kommentare kamen von ARVID REANATE. Dem Autoren GnuPGs, WERNER KOCH, möchte ich außer für jenes Programm auch für seine technischen Anmerkungen und Korrekturen zu diesem Handbuch danken.

△ CHRISTOPHER CREUTZIG; ANDREAS BUHL ist zum Zeitpunkt, da ich dies schreibe, noch in seinem wohlverdienten Urlaub und muß mit allen meinen Textänderungen und auch meinem Vorwort und den von mir geschriebenen Danksagungen leben.

⊖ Diese Anleitung durften wir freundlicherweise auf die beiliegende CD aufnehmen. Sie finden Sie im Verzeichnis `Doku/Raven`.

2. Überblick

2.1. Zusammenfassung

PGP bietet ein System mit öffentlichen Schlüsseln für die Verschlüsselung von E-Mail und von Dateien. Es ermöglicht eine sichere Kommunikation zwischen Personen, die sich nie direkt getroffen haben müssen. Ein abhörsicherer Kanal für den Austausch eines Schlüssels ist nicht erforderlich. PGP bietet viele Möglichkeiten und ist schnell. Es hat eine ausgefeilte Schlüsselverwaltung, bietet digitale Unterschriften, komprimiert die unverschlüsselten Daten und ist auf beinahe jedem Betriebssystem einsetzbar.

2.2. Schnellanleitung

In diesem Abschnitt finden Sie die Schritte, die Sie üblicherweise durchführen, wenn Sie PGP zum ersten mal verwenden. Wir empfehlen Ihnen dringend, das Handbuch wenigstens zu überfliegen – zwingen können und wollen wir Sie natürlich nicht.

1. Installieren Sie PGP. Unter Windows führen Sie dazu das jeweilige Installationsprogramm durch einen Doppelklick aus.
   2.6.x: S. 72, GnuPG: S. 76, 5.0i Kommandozeile: S. 74
   5.0i Windows: S. 148, 5.5.3i Windows: S. 153, 6.0i Windows: S. 160

2. Erzeugen Sie ein Schlüsselpaar. Sie können diesen Schritt bei den Windows-Installationen gleich am Ende automatisiert ausführen lassen; wir raten davon ab, siehe Abschnitt 17.1 auf Seite 146.
   Kommandozeile: S. 80
   Windows: S. 166

3. Machen Sie ein paar Probe-Ver- und Entschlüsselungen an sich selbst. Experimentieren Sie auch mit digitalen Unterschriften.
   Kommandozeile: S. 88, S. 92, S. 130, S. 90, S. 125
   Windows: S. 231, S. 237, S. 234

4. Senden Sie Ihren öffentlichen Schlüssel an andere.

Kommandozeile: S. 85
Windows: S. 178

5. Lesen Sie Ihnen zugesandte Schlüssel Ihrer Bekannten ein oder holen Sie sie vom Keyserver.

Kommandozeile: S. 83, S. 129
Windows: S. 173, S. 199

6. Überprüfen Sie die öffentlichen Schlüssel Ihrer Freunde und Bekannten und unterschreiben Sie sie.

Kommandozeile: S. 60, 86
Windows: S. 60, 179

7. Verschlüsseln und signieren Sie Ihre E-Mail.

Kommandozeile: S. 88, S. 92, S. 130, S. 90, S. 125
Windows: S. 231, S. 237, S. 234, S. 248

2.3. Allgemeines

PGP steht für „Pretty Good™ Privacy“, zu deutsch etwa „recht gute Privatsphäre“, wobei „Pretty Good“ auch als Kurzform von „Phil's Pretty Good Software“ anzusehen ist, von wo das Programm ursprünglich stammt. Es handelt sich um ein hochsicheres Ver- und Entschlüsselungsprogramm, das für sehr viele verschiedene Rechner und Betriebssysteme existiert, so z. B. auf Amiga, Atari, MacIntosh, MS-DOS, Microsoft Windows (3.1, 3.11, 95, 98, NT), Unix, VAX/VMS etc.

GnuPG steht für „Gnu Privacy Guard“, also „Gnu-Wächter für Privatsphäre“, wobei das Akronym „Gnu“ für „Gnu is not Unix“ steht. GnuPG ist eine freie♣ Implementation des OpenPGP-Standards, der aus dem Datenformat von PGP 5.x/6.x entwickelt wurde. Es ist derzeit unter den meisten Unix-Varianten und (mit leichten Einschränkungen) Windows 95/98/NT einsetzbar. PGP und GnuPG gestatten den Austausch von Nachrichten ohne Verzicht auf Privatsphäre, Authentifikation und Komfort.

♣ Um es mit den Worten RICHARD STALLMANS zu sagen: „frei“ bezieht sich auf „Freiheit“, nicht auf den Preis. Nebenbei ist GnuPG auch gratis, aber der englische Begriff „free“ hat im Zusammenhang mit Software *nicht* die Bedeutung „gratis“.

Hierbei verstehen wir unter *Privatsphäre*, daß eine Nachricht nur von den gewünschten Adressaten gelesen werden kann, unter *Authentifikation*, daß eine Nachricht überprüfbarerweise von der Person stammt, von der sie zu stammen scheint (elektronische Unterschrift), und unter *Komfort*, daß der Anwender sich keine unnötigen Gedanken darüber machen muß, wo er welche Schlüssel aufbewahrt und welchen Schlüssel er zum Datenaustausch mit wem benutzt. Dies ist der hauptsächliche Schwachpunkt bei herkömmlicher Verschlüsselungssoftware.

Ebenfalls im Gegensatz zu herkömmlicher Verschlüsselungstechnik werden keine abhörsicheren Kanäle gebraucht, durch die Schlüssel ausgetauscht werden, da PGP ein System mit öffentlichen Schlüsseln nutzt. PGP verbindet die Bedienungsfreundlichkeit der öffentlichen Schlüssel mit der Geschwindigkeit konventioneller Kryptographie und verwendet Verfahren zur Generierung von Textprüfsummen, um elektronische Unterschriften zu erzeugen. Es komprimiert die Daten vor der Verschlüsselung, ist benutzerfreundlich, hat eine komfortable Schlüsselverwaltung und ist portabel. Sie können von Ihrem PC, der beispielsweise unter Windows NT läuft, einen Text verschlüsselt an einen Bekannten senden, der diesen auf seinem Amiga entschlüsseln kann. Oder Sie senden einen Text an die Uni, den Sie für wichtig halten und deshalb unterschreiben möchten – kein Problem; wenn die Uni PGP installiert hat, kann die Empfängerin Ihre Unterschrift auch auf einer Unix-Maschine prüfen. Und PGP ist schnell. PGP ist das Kryptographiesystem für Jedefrau und Jedermann.

PGP versendet seine Dateien nicht selbst, sondern beschäftigt sich ausschließlich mit dem Ver- und Entschlüsseln. Zum Versenden brauchen Sie weitere Programme, und zwar dieselben wie für nicht verschlüsselte Texte.

2.4. Einsatzgebiete PGPs

PGP ist ein Programm für die Verschlüsselung persönlicher Nachrichten. Richtig eingesetzt, schließt es die Möglichkeit, per E-Mail oder Diskette versandte Daten ohne Berechtigung zu entschlüsseln, weitgehend aus. Weiterhin ist es bestens dazu geeignet, Dateien elektronisch zu unterschreiben und sie damit vor unbemerkter Manipulation zu schützen – ganz ähnlich wie das inzwischen für Programme und „active content“ auf WWW-Seiten von etlichen Betriebssystemen unterstützt wird. Es ist aber kaum dazu geeignet, Daten auf einem Computer in komfortabler

Weise vor fremdem Zugriff zu schützen,× das gezielte Ent- und Verschlüsseln einzelner Dateien bei Bedarf ist kompliziert und zeitraubend. Dies sollte bei der Verwendung von PGP beachtet werden.

2.5. Die Funktionsweise PGPs

Das Ganze ist natürlich leichter zu verstehen, wenn Sie bereits etwas von Verschlüsselungssystemen im allgemeinen und asymmetrischen Verfahren (auch bekannt als „public-key-Systeme“)im besonderen verstehen. Da dies nicht vorausgesetzt werden kann, nun eine kleine Einführung:

Nehmen wir einmal an, ich möchte Ihnen eine Nachricht schreiben, die sonst niemand lesen können soll. Hierzu kann ich die Nachricht „verschlüsseln“ oder „codieren“, das heißt, ich verändere sie nach einem ausgeklügelten System, so daß niemand etwas damit anfangen kann, ausgenommen Sie, die beabsichtigte Empfängerin oder der beabsichtigte Empfänger der Nachricht. Ich verwende einen Schlüssel, und Sie müssen diesen Schlüssel ebenfalls benutzen, um die Nachricht zu entschlüsseln. Zumindest bei herkömmlichen symmetrischen, also mit einem einzigen Schlüssel arbeitenden Systemen. Das Ganze funktioniert und ist mit erfreulich bescheidenem Aufwand sehr sicher zu machen – Banken, Geheimdienste, Militärs und andere arbeiten seit Jahren so. Dieser „Schlüssel“ entspricht in etwa dem Schlüssel eines Safes. Das Verschlüsseln einer Nachricht entspricht dem Verschließen des Safes, das Entschlüsseln dem Öffnen. (Die hier genannten Safes lassen sich problemlos jederzeit erzeugen. Verschlossene Safes samt Inhalt zu kopieren, ist ebenfalls kein Problem. Aber das sind nun einmal angenehme Eigenschaften aller elektronischen Daten.)

Daß herkömmliche Systeme einen gemeinsamen Schlüssel zum Ver- und Entschlüsseln benutzen, bedeutet, daß dieser Schlüssel auf einem sicheren Weg zwischen Absenderin und Empfänger ausgetauscht werden muß. Das kann umständlich sein, vor allem, wenn Sie mit einem Menschen Kontakt aufnehmen möchten, den Sie nie getroffen haben. Außerdem: Wenn ein sicherer Weg existiert, um den Schlüssel auszutau-

× Mit PGPDisk stellt PGP 6.0 eine derartige Funktionalität unter Windows zur Verfügung. Dabei handelt es sich aber um eine losgelöste Funktionalität, die nicht zum eigentlichen Aufgabengebiet von PGP gehört und aus diesem Grund auch in ein eigenes Programm ausgelagert ist, das außer dem Namen und der Herstellerfirma nicht viel mit PGP zu tun hat. Eine Besprechung dieses Programms würde den Rahmen dieses Handbuchs sprengen; außerdem gibt es in diesem Bereich viele Programme mit derselben Funktionalität.

schen, warum wollen Sie dann verschlüsseln? Sie könnten den sicheren Kanal doch auch für die Kommunikation nutzen. (Ein oder zwei Ausnahmen gibt es natürlich, z. B. könnten Sie sich treffen und anschließend erst Nachrichten austauschen oder Sie haben zwar einen sicheren Kanal, können aber nur wenige Daten darüber senden. Keinen sicheren Kanal zu benötigen, ist aber zweifelsohne von Vorteil.)

In Systemen mit öffentlichen Schlüsseln (asymmetrische Systeme) gibt es für jeden Teilnehmer ein Schlüsselpaar, bestehend aus einem öffentlichen und einem geheimen Schlüssel. Was mit dem öffentlichen Verschlüsselungsschlüssel verschlüsselt wurde, kann nur mit dem dazugehörigen geheimen Entschlüsselungsschlüssel entschlüsselt werden. Den öffentlichen Schlüssel zu kennen, reicht nicht aus, um die damit verschlüsselten Nachrichten lesen zu können, nicht einmal der Absender selbst kann das. Wenn symmetrische Systeme mit einem Safe verglichen werden, so entsprechen die asymmetrischen Systeme in dieser Hinsicht einem Briefkasten (mit der Sicherheit eines Safes): Jeder kann Nachrichten hineinstecken, aber nur berechtigte Personen können mit ihrem Schlüssel den Briefkasten öffnen und Nachrichten herausnehmen. Der öffentliche Teil des Schlüssels entspricht hier dem Kasten (wiederum in beliebiger Anzahl kopierbar), der private Teil dem Schlüssel, mit dem der Briefkasten sich öffnen läßt. Der geheime Schlüssel$^\diamondsuit$ läßt sich nicht aus dem öffentlichen Schlüssel berechnen. Deshalb kann der öffentliche Schlüssel ohne Bedenken verbreitet werden (darum heißt er „öffentlich“), womit ein sehr viel geringerer Bedarf an sicheren Transportwegen besteht als bei herkömmlichen Systemen.

Da alle bekannten asymmetrischen Algorithmen merklich langsamer arbeiten als herkömmliche Verschlüsselungsalgorithmen, bietet es sich an, die eigentliche Verschlüsselung mit einem guten und schnellen herkömmlichen System vorzunehmen und den dabei verwendeten Schlüssel (was eine verschwindend geringe Datenmenge darstellt) mit einem asymmetrischen System an den Empfänger zu codieren. Diese Mischung aus zwei Verfahren nennt sich „hybrides Verschlüsslungssystem“. PGP generiert für jede Verschlüsselung hierzu einen zufällig ausgewählten Schlüssel, der nur ein einziges Mal verwendet wird. Diesen Schlüssel werden wir im Folgenden „Sitzungs-Schlüssel“ oder „session key“ nennen. Mit diesem Schlüssel verschlüsselt PGP die Nachricht mit

$\diamondsuit$ Wir verwenden die Begriffe „privater Schlüssel“, „geheimer Schlüssel“, „privater Teil des Schlüssels“ und „geheimer Teil des Schlüssels“ als Synonyme; sie haben alle dieselbe Bedeutung.

einem schnellen, guten, symmetrischen Verfahren. Anschließend wird dieser Sitzungs-Schlüssel mit dem öffentlichen Schlüssel des Empfängers codiert und zusammen mit der verschlüsselten Nachricht in eine Datei geschrieben. Der Empfänger kann nun mit Hilfe seines privaten Schlüssels den zufällig gewählten Schlüssel wieder entziffern und die gesamte Nachricht damit lesbar machen. Dieser ganze Vorgang wird dem Benutzer durch die Software abgenommen. Um im Bild zu bleiben: Der Absender verpackt die Nachricht in einen (frisch erzeugten) Safe mit zufällig gewähltem Schlüssel. Diesen Schlüssel steckt er in den Briefkasten des Empfängers und schickt beides auf die Reise. Der Empfänger öffnet den Briefkasten, nimmt den Schlüssel zum Safe heraus und öffnet den Safe.

Weiterhin bieten einige asymmetrische Verfahren die Möglichkeit, eine Nachricht zu „unterschreiben“ („digitale Unterschrift“). Hierzu kann der Absender einer Nachricht diese mit seinem privaten Signaturschlüssel codieren, und jeder Empfänger kann die Echtheit des Absenders mit dessen öffentlichem Signaturschlüssel prüfen. Gelingt dies, so ist die Nachricht mit dem privaten Schlüssel codiert, also unterschrieben worden. Die Absenderangabe ist dann echt. Das läßt sich jetzt nicht mehr gut mit der Briefkastenanalogie erklären; das Vorgehen entspricht mehr einem Schaukasten. Zettel, die im Schaukasten einer bestimmten Organistaion hängen, sind dadurch von dieser Organisation legitimiert, denn ohne den Schlüssel zum Kasten kann niemand Zettel hineinhängen. Der öffentliche Teil des Signaturschlüssels ermöglicht es, den Kasten selbst zu prüfen und festzustellen, wer ihn verschlossen hat.

Nebenbei bemerkt verwendet das Bundesverfassungsgericht PGP, um die auf `www.bundesverfassungsgericht.de` abgelegten Urteile zu signieren. Die entsprechende Signatur steckt in HTML-Kommentaren, Sie können sie also erst sehen, wenn Sie sich den Quelltext der Seite anzeigen lassen. Die Seite als HTML-Datei zu speichern und die Unterschrift zu prüfen, funktioniert aber.

Verschlüsselung und digitale Unterschrift können natürlich miteinander kombiniert werden, um Briefgeheimnis und Authentizität des Absenders zu gewährleisten: Die Nachricht wird zunächst mit dem eigenen privaten Signaturschlüssel signiert und diese unterschriebene Nachricht anschließend mit dem öffentlichen Schlüssel der Empfängerin codiert. Diese decodiert die Nachricht zunächst mit ihrem privaten Schlüssel und prüft anschließend mit dem öffentlichen Signaturschlüssel des Absenders die Unterschrift. PGP erledigt die einzelnen Schritte

automatisch, ohne daß Sie sich um die Einzelheiten kümmern müssen.

Die Analogie wird jetzt ein wenig skurril: Stellen Sie sich die Situation einmal vor. Sie möchten eine Nachricht versenden, die nur ein bestimmter Empfänger lesen kann, bei der aber gesichert ist, daß sie von Ihnen stammt. Dann können Sie diese Nachricht in einen Glaskasten legen, bei dem jeder nachprüfen kann, daß er von Ihnen verschlossen wurde (Unterschrift). Diesen Kasten legen Sie dann in einen zufällig gewählten Safe (konventionelle Verschlüsselung) und den Schlüssel des Safes stecken Sie in eine Art Briefkasten des Empfängers. Anschließend schicken Sie den Briefkasten und den Safe mit dem Glaskasten drin an den Empfänger. Dieser kann nun (als Einziger!) den Briefkasten öffnen, ihm den Schlüssel entnehmen, damit den Safe öffnen und Ihre Nachricht lesen. Außerdem kann er durch den Glaskasten, in dem Sie die Nachricht in den Safe gelegt haben, verifizieren, daß die Nachricht tatsächlich von Ihnen stammt.

Sicherheit dieses Vorgehens

Viele Leute haben sich Gedanken gemacht und gefragt, ob die Verwendung eines symmetrischen Verfahrens die Sicherheit von PGP nicht beeinträchtigt. Dies ist nicht der Fall, denn um asymmetrische Verfahren auf ein Sicherheitsniveau zu bringen, das den in PGP verwendeten symmetrischen Verfahren entspricht, ist ein enormer Aufwand nötig. Der große Vorteil von asymmetrischen Verfahren wie RSA liegt nicht in der Sicherheit, die sie bieten, sondern in der enormen Vereinfachung der Schlüsselübergabe. Die Verwendung symmetrischer Algorithmen für die eigentliche Verschlüsselung schwächt PGP keinesfalls ab. Näheres hierzu finden Sie in Abschnitt 4.4.3 auf Seite 28.

Technische Details

Die Schlüssel werden in sogenannten Schlüsselzertifikaten („key certificates“) aufbewahrt, die außer dem Schlüssel selbst noch einen kurzen Text mit Namen und Netzadresse der „Inhaberin“ und den Vermerk, wann der Schlüssel erzeugt wurde, enthalten. Der Text mit Namen und Netzadresse wird im folgenden „Benutzer-ID“ genannt. „Public key certificates“ enthalten die öffentlichen Schlüssel, während „secret key certificates“ die privaten Schlüssel beinhalten. Private Schlüssel werden

verschlüsselt gespeichert, um sie einem Angreifer, der die Datei kopiert, nicht schutzlos auszuliefern. Genaueres zum Aufbau und der Bedeutung eines Schlüsselzertifikates finden Sie in Abschnitt 4.1 auf Seite 19.

PGP benutzt zwei Dateien, in denen die Zertifikate aufbewahrt werden, eine für die öffentlichen und eine für die privaten. Diese Dateien können Sie mit (je) einem Schlüsselbund vergleichen; deswegen werden wir sie im Folgenden auch „Schlüsselbunde“ nennen.

Die Software der Empfängerin entschlüsselt ankommende Nachrichten automatisch. Am Anfang einer PGP-verschlüsselten Nachricht stehen die Schlüsselkennungen der Empfänger. PGP sucht in der Datei mit privaten Schlüsseln nach einem passenden Schlüssel und entschlüsselt damit die Nachricht. Alles automatisiert.

3. Schwachstellen PGPs

Es gibt keine Sicherheit ohne Schwachstellen. Auch PGP ist davon nicht ausgenommen und bietet einige Ansatzpunkte, wie der Schutz umgangen werden könnte. Die wichtigsten, an die Sie immer denken sollten:

Angriff	besprochen in
Ihr privater Schlüssel fällt in fremde Hände	5.1, S. 34
jemand verfälscht öffentliche Schlüssel	5.2, S. 35
Sie löschen Ihre Dateien nicht gründlich genug	5.4, S. 37
Viren und Trojanische Pferde	5.5, S. 39
unbefugter Zugriff auf Ihren Rechner	5.6, S. 41
elektromagnetische Abstrahlungen	5.7, S. 42
Übergriffe auf Multi-User-Systemen	5.8, S. 43
Überwachung ihres Datenverkehrs	5.9, S. 44
Kryptanalyse	5.10, S. 47

Vertrauen in Placebos und Wundermedikamente?

Wenn wir ein Verschlüsselungsprogramm betrachten, stellt sich die Frage: Warum sollte ich diesem Produkt vertrauen? Auch den Quellcode zu untersuchen hilft nicht viel weiter, denn die meisten Menschen kennen sich in den Grundlagen der Kryptographie nicht genug aus, um die Sicherheit zu beurteilen. Und selbst wenn, können sie immer noch nicht sicher sein, daß keine Hintertür eingebaut ist, die sie eventuell übersehen. (Hierzu ist die Besprechung des ARR-„Features“ auf Seite 267 interessant.)

PHILIP ZIMMERMANN berichtet aus seiner Zeit am College von einem Erlebnis, das ihm klargemacht hat, wozu „Basteleien“ in diesem Bereich allenfalls führen. Er erfand ein (seiner Meinung nach geniales) Verschlüsselungssystem. Es funktionierte so, daß ein Zufallszahlengenerator Zahlen ausspuckte, die zu den zu verschlüsselnden Zeichen

addiert wurden. Somit wäre eine Häufigkeitsanalyse des entstehenden Textes ausgeschlossen, was ein Knacken des Codes unmöglich machen sollte. Einige Jahre später fand er eben dieses Schema in einigen Einführungswerken zur Kryptographie. Die Freude wurde jedoch schnell getrübt, denn es wurde dort als Beispiel für einen leicht knackbaren Code verwendet.⊗

Dieses Beispiel zeigt, wie leicht es ist, einer trügerischen Sicherheit zu verfallen, wenn es um einen neuen Verschlüsselungsalgorithmus geht. Auch wenn die meisten Menschen dies nicht direkt nachvollziehen können, ist es extrem schwer, ein Schema zur Verschlüsselung zu entwickeln, das einem ernstgemeinten und mit entsprechendem Hintergrund durchgeführten Angriff standhält. Auch viele kommerzielle Produkte bieten – aufgrund der verwendeten Rechenvorschriften – keine ernstzunehmende Sicherheit. Gerade in punkto Sicherheit wird sehr viel minderwertige Ware verkauft.

Stellen Sie sich vor, Sie kaufen ein neues Auto, und eine Woche später sehen Sie die Aufzeichnung eines Crash-Tests, in dem die wunderschönen Sicherheitsgurte einfach reißen. Da *kann* es besser sein, gar keine Sicherheitsgurte zu haben, da Sie sich ansonsten in falscher Sicherheit wiegen. Dasselbe gilt für Software – wenn Sie sich auf schlechte Software verlassen, um die Vertraulichkeit Ihrer Daten zu gewährleisten, können Sie eine extrem böse Überraschung erleben. Oder – noch schlimmer – eventuell bemerken Sie nicht einmal, daß Ihre Daten von Unbefugten gelesen wurden, bis es zu spät ist. War Ihnen übrigens klar, daß Sicherheitsgurte nach jedem Unfall, soätestens aber nach zehn Jahren, gewechselt werden müssen, um Sicherheit zu gewährleisten?

Aber auch die Verwendung bewährter Algorithmen bietet keine Sicherheit, wenn sie nicht kompetent eingesetzt werden. So empfiehlt beispielsweise die Regierung der USA die Verwendung des Federal Data Encryption Standard, DES,⊙ allerdings nur für kommerzielle Anwendungen – Informationen unter staatlicher Geheimhaltung dürfen damit nicht verschlüsselt werden. Aber das ist ein anderes Thema. Bei DES

⊗ Der Vollständigkeit halber sei erwähnt, daß dieser Algorithmus durchaus sicher sein kann – es hängt nur davon ab, welcher „Zufallszahlen"-Generator verwendet wird. Die so erzeugte Gruppe von Verschlüsselungssystemen sind die „additiven Stromchiffren".

⊙ Zum Zeitpunkt der Drucklegung dieses Buches ist das mit Einschränkungen noch richtig. Das Auswahlverfahren für den Nachfolgealgorithmus läuft derzeit. Für das, was in diesem und dem nächsten Absatz steht, ändert sich aber vermutlich leider nichts, wenn der AES (Advanced Encryption Standard) verabschiedet wird.

gibt es verschiedene Stufen von Sicherheit. Die schwächste, von der die US-Regierung abrät, ist die sogenannte ECB-Verschlüsselung (Electronic Codebook). Besser sind Cipher Feedback (CFB) oder auch Cipher Block Chaining (CBC).

Leider verwenden die meisten kommerziellen Produkte, die auf DES basieren, die ECB-Methode. In Gesprächen mit Autoren solcher Software stellt sich oft heraus, daß sie nie etwas von CBC oder CFB gehört haben, nicht einmal von möglichen Schwachstellen des ECB. Dabei sind die Programme, auf die Sie sich am wenigsten verlassen sollten, immer noch die, bei denen der Programmierer verschweigt, wie die Verschlüsselung funktioniert. Um fair zu bleiben, muß aber betont werden, daß diese Produkte normalerweise nicht von Firmen kommen, die sich auf Kryptographie spezialisiert haben.

Und falls Sie jetzt noch der eingebauten Verschlüsselung von WordPerfect, Lotus 1-2-3, MS Excel, Symphony, Quattro Pro, Paradox oder MS Word vertrauen – wenden Sie sich an die Firma AccessData (87 East 600 South, Orem, Utah 84058, USA, `http://www.accessdata.com/`), dort können Sie für 158 US-Dollar ein Softwarepaket erhalten, das eben diese Systeme entschlüsselt. Gekauft wird dieses Programm von Leuten, die ihr Paßwort vergessen haben, und von Strafverfolgungsbehörden. Der Autor des Programms, ERIC THOMPSON, sagte übrigens, er habe einige Verzögerungsschleifen eingebaut, damit das Knacken des Paßwortes nicht so einfach aussieht wie es ist. Auch die `PKzip`-Verschlüsselung ist seiner Aussage nach einfach zu umgehen.

Verschlüsselungssoftware läßt sich mit Medikamenten vergleichen. In beiden Fällen kann die Wirksamkeit von größter Bedeutung sein. Ebenso wie Penizillin sieht man es einer Verschlüsselungssoftware nicht an, ob sie gut arbeitet. Jeder kann feststellen, ob sein Textverarbeitungssystem gute Arbeit leistet,[∨] aber woran erkennt der durchschnittliche Anwender, ob seine kryptographische Software gut verschlüsselte Dateien liefert? Ein Laie kann den Unterschied zwischen schlecht oder gut verschlüsselten Daten nicht erkennen.

Deshalb gibt es auch so eine Vielzahl schlechter Verschlüsselungsprogramme. Die „Verschlüsselung“, die viele Programme quasi nebenbei anbieten, ist im Allgemeinen noch viel schlechter. Bei Verschlüsselungsprogrammen gibt es viel Pfusch. Aber im Gegensatz zu den Leuten,

∨ Die massenhafte Verwendung einer ausgesprochen schlechten Textverarbeitungssoftware scheint dem zu widersprechen, aber sei's drum.

die Patentmedizin verhökern, wissen viele Programmierer offensichtlich nicht einmal, daß sie Quacksalberei betreiben. Diese Programmierer sind oft dennoch fähige Leute, aber die wenigsten haben auch nur ein einziges wissenschaftliches Buch über Kryptographie gelesen. Trotzdem glauben sie, sie könnten gute Verschlüsselungsprogramme schreiben. Und warum auch nicht? Verschlüsselung scheint zunächst einmal einfach machbar zu sein. Und die Programme *scheinen* auch ganz ordentlich zu arbeiten.

Jeder, der glaubt, er habe ein unknackbares Verschlüsselungsverfahren entwickelt, ist entweder ein unglaublich seltenes Genie, oder er ist naiv und unerfahren. Leider hat noch niemand einen Crash-Test für Verschlüsselungsalgorithmen erfunden – einen Code zu analysieren, ist eine recht zeitaufwendige Sache, die sich nicht automatisieren läßt und die im Allgemeinen erst dann lohnt, wenn ein Verfahren eine gewisse Verbreitung genießt. Aus diesem Dilemma gibt es einen Ausweg, nämlich den, nur analysierte und für gut befundene Verfahren einzusetzen. Glücklicherweise gibt es genügend davon.

BRIAN SNOW, ein hochrangiger Kryptograph der NSA, eines der US-amerikanischen Geheimdienste, sagte einmal, er würde keinem Verschlüsselungsalgorithmus über den Weg trauen, der von jemandem entwickelt sei, der nicht sehr viel Erfahrung mit dem Knacken von Verschlüsselungen hat. Eine durchaus sinnvolle Einstellung. Im Bereich der kommerziellen Softwareentwicklung scheint es fast niemanden zu geben, auf den dieses Kriterium zutrifft. SNOW sieht das ähnlich: „Und das macht unseren Job bei der NSA um einiges einfacher.“ Gruselige Vorstellung.

Auch die US-Regierung hat Wundermedizin verbreitet. Nach dem zweiten Weltkrieg verkaufte sie beispielsweise Enigma-Maschinen (die Verschlüsselungsgeräte der deutschen Wehrmacht im zweiten Weltkrieg) an Regierungen von Entwicklungsländern, ohne diesen zu sagen, daß diese Verschlüsselung während des Krieges von polnischen und britischen Mathematikern geknackt worden war.⊕ RIVEST, SHAMIR und ADLEMAN haben 1977 den RSA-Algorithmus veröffentlicht, ohne zuvor zu versuchen, ihn zu patentieren, weil sie befürchten mußten, daß die US-Regierung ansonsten versucht hätte, die Veröffentlichung zu ver-

⊕ Diese Arbeiten in England waren einer der wichtigsten Anstöße für die Entwicklung dessen, was heute die Informatik ist. Einer der Mathematiker dort hieß ALAN TURING, mit diesem Namen sind bis heute einige der wichtigsten Erkenntnisse der theoretischen Informatik verbunden.

hindern. (Das ist auch der Grund dafür, weshalb RSA nur in den USA patentiert ist. Das US-amerikanische Patentrecht gestattet es Erfindern, auch nach der Veröffentlichung ihrer Resultate einen Patentantrag einzureichen.) Auch die Entwicklung abhörsicherer Telephontechnik für die Allgemeinheit wurde durch die US-Regierung behindert.

Wir möchten dieses Kapitel mit einigen Worten PHILIP ZIMMERMANNs aus der Anleitung zu PGP 2.6.2 beenden: „Ich bin von der Sicherheit von PGP nicht so überzeugt, wie während meines Studiums von der Sicherheit meines „genialen“ Verschlüsselungsverfahrens. Wäre ich von PGP vollkommen überzeugt, wäre das ein schlechtes Zeichen. Aber ich bin ziemlich sicher, daß PGP keine ins Auge springenden Schwachstellen hat. Die Algorithmen, die in PGP verwendet werden, stammen von zivilen Kryptographen mit sehr gutem Ruf, und sie sind eingehend untersucht worden. Selbstverständlich ist der komplette Sourcecode erhältlich, so daß jeder, der programmieren kann (oder eine vertrauenswürdige Bekannte hat, die dazu in der Lage ist), das System durchleuchten kann. Der Quellcode ist über Jahre professionell entwickelt worden. Im übrigen arbeite ich nicht für die NSA. Ich hoffe, daß der Schritt, Vertrauen in PGP zu gewinnen, nicht zu viel Überwindung kostet.“

4. Ein Blick auf's Detail

4.1. Schlüsselzertifikate

Wie in Abschnitt 2.5 bereits angesprochen, werden die öffentlichen Schlüssel nicht als „nackte Daten“ ausgetauscht, sondern enthalten Zusatzinformationen, um sie Personen und Einsatzzwecken zuordnen zu können, um ihre Echtheit und Gültigkeit feststellen zu können und dergleichen mehr. PGP 2.6.x benutzt das Datenformat, wie es in RfC 1991 beschrieben ist, PGP 5.x/6.x und GnuPG benutzen das Datenformat wie in RfC 2440 („OpenPGP“) definiert.[∞] Wir bezeichnen das Format nach RfC 1991 als „altes Format“ und RfC 2440 (genauer: die dort als V4 bezeichnete Version eines Schlüsselzertifikats) als „neues Format“. Private Schlüssel haben ein eigenes Format für Zertifikate, das sich vom Format für öffentliche Schlüssel aber kaum unterscheidet und nur sehr selten gebraucht wird (private Schlüssel werden nicht so oft weitergegeben wie öffentliche . . .), weswegen wir es hier nicht näher betrachten werden.

Ein Zertifikat enthält zunächst einmal natürlich den öffentlichen Schlüssel selbst; dieser Eintrag besteht aus mehreren langen Zahlen. Genaueres zu diesen Zahlen finden Sie in der Beschreibung der jeweiligen Algorithmen. Hinzu kommt ein Texteintrag, der angibt, wem dieser Schlüssel gehört. Im alten Datenformat ist hierfür ein einzelnes Textfeld vorgesehen, innerhalb dieses Feldes sollten Sie aus Portabilitätsgründen nur ASCII verwenden, also z. B. keine Umlaute. Im neuen Datenformat hingegen können Sie (prinzipiell) im Namen beliebige (Unicode-)Zeichen verwenden.[⋆]

∞ Jedes der genannten Programme hat oder ermöglicht leichte Abweichungen vom RfC 2440.

⋆ Meine Versuche führten so weit, daß ich einen Schlüssel mit Umlauten, japanischen Zeichen etc. erzeugen konnte und ihn auch angezeigt bekam – mit GnuPG; PGP 5.0 zeigte ein ‚ü‘ als \374 an, außerdem fehlten die letzten Zeichen des Namens. Einen Schlüssel mit Umlauten anzuwählen, funktionierte wiederum nur mit PGP 5.0; dieser Fehler dürfte in GnuPG aber auch in absehbarer Zeit behoben werden.

Im neuen Datenformat haben Sie die Möglichkeit, innerhalb eines Zertifikates mehrere Schlüssel (mit einer gemeinsamen User-Kennung) zu haben. Es ist beispielsweise sinnvoll, einen lange gültigen Schlüssel zu verwenden, um andere Schlüssel zu unterschreiben (Master-Key) und Schlüssel mit kürzerer Gültigkeitsdauer für die Verschlüsselung von Nachrichten an Sie oder für digitale Unterschriften unter Texte und Dateien. Sie können natürlich auch verschiedene Schlüssel für verschiedene Einsatzmöglichkeiten verwenden. Beispielsweise ist es sinnvoll, einen anderen Schlüssel für Firmenpost zu verwenden als für private Mails – den Firmenschlüssel können Sie dann auch einer Urlaubs- oder Krankheitsvertretung geben, wenn es nötig wird, ohne daß diese gleich Ihre private Post lesen kann. Über diese Möglichkeit ist auch realisiert, daß jetzt verschiedene Algorithmen (und damit auch zwangsweise verschiedene Schlüssel) für Unterschriften und Verschlüsselung verwendet werden können, beispielsweise die Mischung DSA/ElGamal.

4.2. Schlüsselkennungen

Jeder PGP-Schlüssel hat zusätzlich zum Namen und der E-Mail-Adresse eine Kennung, die aus den vom Programm generierten Schlüsseldaten abgeleitet wird. Diese Kennung verwendet PGP intern (d. h., ohne den Benutzer damit zu behelligen), um die Schlüssel voneinander zu unterscheiden. Zwei verschiedene Schlüssel können dieselbe Benutzer-ID haben, die Kennung aber ist aller Wahrscheinlichkeit nach stets verschieden.

PGP 2.6.x verwendet die letzten 64 Bit des Schlüssels,⊘ von denen nur die letzten 32 Bit angezeigt werden, z. B. `0x6ce93239`. Mit ein wenig Mathematik ist es ohne nennenswerten Aufwand möglich, zu einer gegebenen Schlüssel-ID (deren letzte Ziffer ungerade ist) einen funktionsfähigen Schlüssel zu erzeugen, die ID hat von daher keinerlei Bedeutung als Sicherheitsmerkmal. (Die Version 2.6.3in hat übrigens auch eine Option, mit der sich derartige Schlüssel erzeugen lassen. Von der Verwendung ist abzuraten. Das Einlesen derartiger Schlüssel kann die Schlüsselverwaltung anderer PGP-Versionen durcheinanderbringen – die ID wird verwendet, um auf einen Schlüssel zuzugreifen.) Nachfolgende Versionen (PGP 5.x/6.x, OpenPGP, GnuPG) verwenden daher als Schlüsselkennung

⊘ Ganz genau: Die letzten 64 Bit des RSA-Modulus.

den Fingerabdruck des Schlüssels (näheres zum Begriff des Fingerabdrucks finden Sie im Abschnitt 4.6).

4.3. Zufallszahlen

PGP und GnuPG benötigen für einige Arbeitsschritte Zufallszahlen, an deren Unvorhersagbarkeit letztlich die Sicherheit der gesamten Verschlüsselung hängt. Zu diesen Schritten gehören das Erzeugen eines Schlüsselpaares, die Wahl eines session key und aus technischen Gründen auch die asymmetrische Verschlüsselung oder das Unterschreiben.

GnuPG arbeitet mit einer externen Quelle für alle benötigten Zufallszahlen, da es hauptsächlich für Systeme konzipiert ist, die eine derartige Quelle bereits zur Verfügung stellen. Linux und FreeBSD erzeugen aus „Umgebungslärm“ wie zum Beispiel minimalen Schwankungen in Festplattengeschwindigkeiten, Mausbewegungen und präzisen Zeitmessungen von Tastatureingaben zuverlässige Zufallszahlen. Übrigens basiert die Implementation der Linux-Zufallszahlen-Quelle `/dev/random` ursprünglich auf PGP-Code. Für den Einsatz unter Windows verwendet GnuPG eine mitgelieferte DLL, die `entropy.dll`, die im Wesentlichen dasselbe leistet. Für den Einsatz auf Unix-Versionen ohne `/dev/random` oder etwas vergleichbares bietet sich der Einsatz eines Perl-Daemons namens „entropy gathering daemon“ an. Auch dieses Programm sammelt Umgebungslärm, um ihn Programmen bei Bedarf als zuverlässige Zufallszahlen zur Verfügung zu stellen.

PGP verwendet einen kryptographisch zuverlässigen Pseudozufallszahlengenerator, um wechselnde Einmal-Schlüssel für die konventionelle Verschlüsselung einzelner Dateien zu erzeugen. Die Datei, die den Startwert für den Zufallszahlengenerator enthält, heißt (in der Standardeinstellung, von der wir im weiteren ausgehen) `randseed.bin`. Sie sollte ebenso wie die anderen von PGP benötigten Dateien in dem Verzeichnis stehen, das durch die Umgebungsvariable `PGPPATH` angegeben wird. Falls die Datei nicht vorhanden ist, wird sie automatisch erzeugt. Sie erhält einen Startwert aus echten Zufallszahlen, die aus dem zeitlichen Abstand von Tastatureingaben u. ä. abgeleitet werden. In die Datei `randseed.bin` werden bei jedem Aufruf des Zufallszahlengenerators neue Daten geschrieben, unter Einbeziehung der aktuellen Uhrzeit in Millisekunden und anderer echter Zufallsdaten.

Die Datei `randseed.bin` sollte wenigstens etwas geschützt sein, um das Risiko klein zu halten, daß ein Angreifer die nächsten Schlüssel, die PGP generieren wird, oder die letzten Schlüssel, die PGP generiert hat, berechnet. Dieser Angreifer hätte Schwerstarbeit zu erledigen, weil PGP die Datei `randseed.bin` vor und nach jeder Benutzung kryptographisch „in die Mangel nimmt". Trotzdem ist es keine unangebrachte Vorsicht, darauf zu achten, daß die Datei nicht in die falschen Hände gerät.

Leser, denen diese algorithmisch abgeleiteten Zufallszahlen unheimlich sind, sollten nicht vergessen, daß sie der Sicherheit derselben konventionellen Verschlüsselungsalgorithmen vertrauen, um Nachrichten zu verschlüsseln. Wenn der Algorithmus für die Verschlüsselung sicher genug ist, sollte er hinreichend zuverlässig sein, um Zufallszahlen zu erzeugen, die den konventionellen Schlüssel bilden. Zu beachten ist noch, daß PGP zur Erzeugung eines Paares von öffentlichem und geheimem Schlüssel, die über längere Zeit sicher sein sollen, echte Zufallszahlen verwendet, die im wesentlichen aus den Zeitabständen von Tastatureingaben abgeleitet werden.

Die Erzeugung von Pseudozufallszahlen, also von Zahlenfolgen, die zwar „zufällig aussehen", die aber aus einem Algorithmus abgeleitet werden, ist eine schwierige Aufgabe. Wegen der „guten Zufallsqualität" wird auch bei Anwendungen, die nichts mit Verschlüsselung zu tun haben, wie Statistik oder numerischer Mathematik, gerne ein Verschlüsselungsalgorithmus verwendet, um „Zufallszahlen" zu erzeugen.+ Die Probleme bei Verschlüsselung und bei der Erzeugung von Zufallszahlen sind ähnlich: In beiden Fällen geht es darum, Bitfolgen zu erzeugen, die möglichst wenig Systematik zeigen. Bei der Verschlüsselung sind diese Bitfolgen der verschlüsselte Text, bei der Erzeugung von Zufallszahlen sind die Bitfolgen eben die Zufallszahlen. Leser, denen die Verwendung der Datei `randseed.bin` trotz dieser Argumente unheimlich bleibt, können sie immer noch vor jedem Start von PGP einfach löschen. Allerdings müssen sie dann für die Verschlüsselung eines Klartextes jedesmal ungefähr 90 Tastendrücke für die Erzeugung einer echten Zufallszahl ausführen. Hierbei sammelt PGP deutlich mehr Informationen als benötigt und speichert das „mehr" in der `randseed.bin`. Normalerweise sind neue Tastatureingaben nur nötig, wenn PGP die Zufallsinformationen in der `randseed.bin` „aufgebraucht" hat.

+ Echte Zufallszahlen kann ein Computerprogramm nicht erzeugen, ohne auf Hardware-Unterstützung oder Faktoren wie Timing von Tastatureingaben und Mausbewegungen zurückzugreifen.

Eine Kleinigkeit ist zu den von PGP verwendeten Zufallszahlen noch anzumerken: Seit der Version 5.0 werden Nachrichten, die aufgrund ihrer Größe in mehrere Teile geteilt werden (vgl. Abschnitt ARMORLINES in 14 auf Seite 113), mit einer Seriennummer versehen, um sicherzustellen, daß die zusammengefügten Teile beim Empfänger auch alle Teile derselben Nachricht und in der richtigen Reihenfolge sind. Bei PGP 5.0 war dieses Feature ein wenig unüberlegt implementiert worden, dort wurde diese Seriennummer aus demselben Pool für Zufallszahlen gewonnen wie die Einmal-Schlüssel. Somit besteht ein – wenn auch unscheinbarer – Zusammenhang zwischen der für jedermann sichtbaren Seriennummer und dem geheimen Einmal-Schlüssel. Das Ganze ist kein besonders großes Sicherheitsloch (niemand hat eine Methode beschrieben, um es auszunutzen), zeigt aber, daß auch Firmen und Personen mit einem guten Hintergrund in Verschlüsselungstechnik nicht vor versehentlich eingebauten subtilen Schwachstellen gefeit sind.

4.4. Die von PGP verwendeten Verschlüsselungsalgorithmen

Wie in Abschnitt 2.5 bereits erwähnt, verwendet PGP eine Kombination aus einem konventionellen (symmetrischen) Verschlüsselungsalgorithmus und einem asymmetrischen Algorithmus. Der mit öffentlichen Schlüsseln arbeitende Algorithmus wird nur dazu verwendet, den für eine einzelne Nachricht verwendeten konventionellen Schlüssel zu chiffrieren, so daß er gemeinsam mit der verschlüsselten Nachricht verschickt werden kann. Im folgenden werden wir zunächst die konventionellen und anschließend die asymmetrischen Algorithmen erläutern.

4.4.1. Die symmetrischen Verfahren

4.4.1.1. IDEA

PGP 2.6.x kennt nur ein symmetrisches Verfahren: IDEA™. IDEA ist ein Algorithmus, der 64 Bit lange Daten-„Blöcke“ mit einem 128 Bit-Schlüssel kodiert. Zur Verdeutlichung, was „128 Bit Schlüssel“ bedeuten: Die 56 Bit, die DES verwendet (was seit einiger Zeit auch Ausfuhrgenehmigungen aus den USA erhält), sind in wenigen Stunden durch einfaches Ausprobieren zu knacken, wenn man einen Teil des Klartextes

kennt oder zumindest erkennen kann (Text, Programm, Bilddatei, ...). Die EFF (electronic frontier foundation, `http://www.eff.org/`) hat die Machbarkeit 1998 demonstriert, indem sie ein Gerät gebaut haben, das eine DES-Verschlüsselung in maximal sieben Stunden auflöst. Mit einer vergleichbaren Hardware für IDEA bräuchte ein Angriff etwa $7 * 2^{128-56} = 33\,056\,565\,380\,087\,516\,495\,872$ Stunden, was etwa $4 * 10^{18}$ Jahren entspricht. Zum Vergleich: Das Alter des Weltalls wird derzeit auf etwa $1.5 * 10^{9}$ Jahre geschätzt, also etwa ein Milliardstel der Zeit, um mit aktueller, speziell optimierter Hardware eine 128 Bit-Verschlüsselung zu brechen. Auch wenn wir eine nochmals deutlich schnellere Entwicklung der Rechner annehmen als bislang geschehen, wird von uns niemand mehr erleben, daß es Hardware nach bisheriger Technologie gibt, die alle möglichen IDEA-Schlüssel ausprobieren kann. Quantencomputer sind ein ganz anderes Thema, aber die müßten erst einmal mehr als fünf bis zehn Bit Rechenleistung aufbringen ...

Im Vergleich mit DES steht IDEA von der Sicht des Anwenders her gut da: Es verwendet längere Schlüssel als DES und ist trotzdem deutlich schneller (DES ist ein Hardware-Standard, IDEA ist auf Software optimiert). Nachteile von IDEA sind, daß es patentiert ist und die Verwendung außer im rein privaten Bereich Lizenzgebühren kostet, und zwar auch in Europa (die Gebühren sind aber sehr niedrig, näheres erfahren Sie bei der Firma Ascom Tech, `http://www.ascom.ch`) und daß IDEA nicht wie DES für sich beanspruchen kann, der mit Abstand am besten analysierte Verschlüsselungsalgorithmus in der öffentlich zugänglichen Literatur zu sein. Aber IDEA hat durchaus auch eine längere Liste von Artikeln über gescheiterte Angriffe vorzuweisen. Bis heute hat IDEA kryptanalytischen Angriffen wesentlich besser standgehalten als andere Verfahren wie FEAL, REDOC-II, LOKI, Snefru und Khafre. IDEA widersteht dem sehr erfolgreichen differentiellen kryptanalytischen Angriff von BIHAM und SHAMIR wesentlich besser als DES. Biham und Shamir untersuchten IDEA erfolglos auf Schwachstellen. Akademische Arbeitsgruppen von Kryptanalytikern aus Belgien, England und Deutschland suchen Angriffsmöglichkeiten bei IDEA, ebenso militärische Geheimdienste mehrerer europäischer Länder. (Letztere veröffentlichen ihre Ergebnisse normalerweise natürlich nicht.) Je mehr und je länger dieser Algorithmus Angriffsversuche aus den gefürchtetsten Arbeitsgruppen der kryptanalytischen Welt auf sich zieht, desto mehr steigt das Vertrauen in ihn.

Ein paar sehr technische Randbemerkungen: Ähnlich wie DES kann IDEA für Cipher Feedback (CFB, Rückführung der verschlüsselten Textes) und für Cipher Block Chaining (CBC, Verkettung von Blöcken verschlüsselten Textes) verwendet werden. Bei PGP wird IDEA mit 64-Bit CFB verwendet. Wenn Ihnen die Unterschiede zwischen ECB, CFB und CBC nichts sagen, ist das nicht wirklich tragisch, solange Sie nicht versuchen, Kryptographie zu implementieren. Für die wirklich Interessierten hier eine kurze Erläuterung: Im einfachsten Betriebsmodus, Electronic Codebook (ECB), wird derselbe Klartextblock immer auf denselben Output abgebildet. Das hat die gravierenden Nachteile, daß dieser Klartextblock erstens wiedererkannt werden kann und es zweitens möglich ist, Nachrichten zu verändern, ohne die Verschlüsselung umgehen zu müssen. CBC und CFB umgehen diese Probleme dadurch, daß der Klartext vor der Verschlüsselung mit dem vorherigen verschlüsselten Block verknüpft wird, also die Verschlüsselung sowohl vom Klartextblock als auch der Nachricht vor diesem Block abhängt. Auch die anderen symmetrischen Verfahren in den neueren PGP-Versionen (in OpenPGP sind 3DES, IDEA, CAST5 und Blowfish definiert) verwenden CFB. Exemplarisch für die Funktionsweise eines Blockchiffre finden Sie eine detaillierte Beschreibung des IDEA-Algorithmus im Anhang D.1.

4.4.1.2. 3DES

Wie im vorangegangenen Kapitel angesprochen, ist eines der Hauptprobleme des DES die kurze Schlüssellänge. Um dieses Problem zu beheben, wäre es auf den ersten Blick logisch, DES zweimal nacheinander mit verschiedenen Schlüsseln anzuwenden. Wie MARTIN HELLMAN (einer der beiden Mathematiker, die 1976 auf die Idee der asymmetrischen Verschlüsselung kamen) zeigen konnte, gewinnt man dadurch aber so gut wie keine Sicherheit – und zwar unabhängig davon, welches Verschlüsselungsverfahren man in dieser Art zu verstärken versucht. Erst mit einer dreimaligen Anwendung von DES läßt sich die effektive Schlüssellänge auf 112 Bit bringen. Dieses Verfahren wird triple-DES oder kurz 3DES genannt und ist der standardmäßig eingesetzte symmetrische Algorithmus in PGP 5.x/6.x.

Ein paar technische Details zu 3DES: Es handelt sich um einen symmetrischen Algorithmus, der Blöcke von 64 Bit unter Verwendung eines 112 Bit langen Schlüssels in Blöcke von wiederum 64 Bit abbildet. 3DES ist ein sogenanntes Feistel-Netzwerk. Für uns ist die wichtigste Ei-

genschaft dieser Gruppe von Verschlüsselungsalgorithmen, daß die Entschlüsselung – ebenso wie bei IDEA – mit demselben Algorithmus geschehen kann, indem die intern generierten Teilschlüssel in umgekehrter Reihenfolge verwendet werden. 3DES kann in allen Block-Chiffre-Modi verwendet werden, PGP verwendet den CFB-Modus.

Die Argumente, die gegen die Verwendung von DES sprechen, beziehen sich meistens auf die Schlüssellänge, was mit 3DES kein nennenswertes Problem mehr darstellt, oder sind eher gefühlsmäßig begründet – viele Designentscheidungen, die in die Entwicklung des Algorithmus eingeflossen sind, sind bis heute nicht erklärt worden und viele Menschen sind nach wie vor davon überzeugt, daß die NSA in der Entwicklung ihre Finger drin hatte und den Algorithmus auf eine ausgeklügelte Art und Weise mit einer Hintertür versehen hat. Fünfzehn Jahre intensiven Studiums durch brillante Wissenschaftler und Horden von Hobby-Kryptographen haben keine Anzeichen für eine Hintertür entdecken können, aber das mulmige Gefühl ob der unverstandenen Teile bleibt dennoch bestehen.

Vom pragmatischen Standpunkt aus ist das einzige gute und begründete Argument, das gegen die Verwendung von 3DES spricht, die niedrige Geschwindigkeit. Wie im vorigen Kapitel bereits erwähnt, ist DES in Software deutlich langsamer als IDEA, und einen Block dreimal zu verschlüsseln, trägt auch nicht gerade zu einer Steigerung der Geschwindigkeit bei. Auf der Haben-Seite ist zu 3DES zu sagen, daß es die (auf der Anzahl der gescheiterten Angriffe basierenden) Vertrauenswürdigkeit von DES übernimmt, dabei aber (nach derzeitigem Kenntnisstand) den größten Nachteil von DES, die kurze Schlüssellänge, vermeidet. Gegenüber IDEA hat 3DES den großen Vorzug, frei von Lizenzgebühren verwendet werden zu können.

4.4.2. Die asymmetrischen Verfahren

4.4.2.1. RSA

RSA (nach den Anfangsbuchstaben von RIVEST, SHAMIR und ADLEMAN) war das erste publizierte Verfahren, das die 1976 veröffentlichte Idee der *asymmetrischen Verschlüsselung* tatsächlich umsetzen konnte. Bis heute ist es noch nicht gelungen, ein Verfahren zu finden, mit dem sich eine RSA-Verschlüsselung schneller brechen läßt als durch Faktorisieren einer großen Zahl in ihre Primfaktoren – und auch dafür ist noch kein schneller Algorithmus gefunden worden.

RSA läßt sich mit denselben Schlüsseln sowohl für Verschlüsselung als auch für elektronische Unterschriften einsetzen und kann auch über elliptischen Kurven eingesetzt werden. Nach Meinung der Befürworter dieser Vorgehensweise erhöht sich die Sicherheit bei gleicher Schlüssellänge beträchtlich. Konservativere Stimmen halten dagegen, daß elliptische Kurven erst wesentlich kürzer untersucht werden als die üblichen Strukturen bei RSA und ElGamal und elliptische Kurven nur dort eingesetzt werden sollten, wo der Platz wirklich knapp ist, beispielsweise bei Chipkarten. RSA ist das bekannteste und am besten untersuchte asymmetrische Verfahren. Der größte Nachteil von RSA ist es, daß es in den USA patentiert ist und seine Verwendung dort deshalb entweder Lizenzgebühren kostet oder (und das nur im nichtkommerziellen Einsatz) eine bestimmte Implementierung namens RSAREF verwendet werden muß, die nur beschränkt lange Schlüssel verwenden kann. Das RSA-Patent in den USA wird am 20. September 2000 auslaufen.

4.4.2.2. ElGamal

Das Verfahren nach ELGAMAL ist eine Modifikation des ersten asymmetrischen Verfahrens, das 1976 von DIFFIE und HELLMAN veröffentlicht wurde. Das Verfahren nach Diffie-Hellman eignet sich ausschließlich dazu, über einen öffentlichen, bidirektionalen Kommunikationsweg (also beispielsweise eine Internet-Verbindung oder per Telephon) einen geheimen Schlüssel zu vereinbaren, ist aber für E-Mail nicht geeignet. Die Verwandschaft führte aber immerhin dazu, daß ElGamal-Schlüssel in PGP 5.x/6.x (fälschlicherweise) Diffie-Hellman-Schlüssel (DH) genannt werden. ElGamal kann für Verschlüsselung und für digitale Unterschriften eingesetzt werden; PGP 5.x/6.x erlaubt den Einsatz für Verschlüsselung, GnuPG beide Einsatzmöglichkeiten. Zur Verwendung ElGamals über elliptischen Kurven gelten die im vorigen Abschnitt angestellten Überlegungen. Nebenbei bemerkt meinen die meisten Leute, die einfach nur von „elliptischen Kurven“ oder „elliptic curve cryptosystems“ reden, ElGamal über diesen Gebilden.

Das Verfahren ist nicht patentiert. Eine Firma namens PKP war der Ansicht, es falle unter das Patent für Diffie-Hellman (auf das sie die Verwertungsrechte hatten). Da jenes Patent 1997 ausgelaufen ist, ist die Frage kaum noch von Interesse.

4.4.3. Warum ein hybrides Verfahren?

Die ausschließliche Verwendung asymmetrischer Verfahren mit langen Schlüsseln ist wegen der langen Rechenzeit für die Ver- und Entschlüsselung großer Datenmengen nicht wirklich brauchbar. Das macht absolut niemand im wirklichen Leben. Trotzdem liegt die Frage nahe, ob die Kombination einer Verschlüsselung mit öffentlichen Schlüsseln und einer zweiten, konventionell arbeitenden Verschlüsselung die Gesamtsicherheit herabsetzt, und das nur, um das Programm schneller zu machen. Schließlich ist eine Kette nur so stark wie ihr schwächstes Glied. Viele Leute, die wenig Erfahrung mit Kryptographie haben, sind der Meinung, daß RSA oder ElGamal vom Prinzip her sicherer sei als eine konventionelle Verschlüsselung.

Das stimmt nicht. Ein asymmetrisches Verfahren kann durch zu kurze oder auch „weiche" Schlüssel leicht angreifbar werden,◁ andererseits können konventionelle Verschlüsselungen bei Wahl eines guten Algorithmus sehr sicher sein.⊤ In den meisten Fällen ist es schwierig, genau zu sagen, wie gut eine konventionelle Verschlüsselung ist, ohne sie wirklich zu knacken. (Dann ist es natürlich leicht zu sagen „höchstens so und so gut".) Ein guter konventioneller Algorithmus kann durchaus schwerer zu knacken sein als ein RSA-Schlüssel der Größenordnung, die PGP als „militärischen Standard" bezeichnet. Der Reiz einer Verschlüsselung mit öffentlichen Schlüsseln besteht nicht darin, daß sie aus sich heraus sicherer als eine konventionelle Verschlüsselung ist – ihr Vorteil besteht in der wesentlich bequemeren und vielseitigeren Handhabung der Schlüssel.

Leute, die beruflich mit der Erforschung von Faktorisierungsalgorithmen zu tun haben, sagen, daß ein Durchprobieren der 2^{128} möglichen Schlüssel für IDEA vom Rechenaufwand her der Faktorisierung eines RSA-Schlüssels mit 3100 Bit entspreche. Das ist deutlich mehr als die 1024 Bit, die die in der US-Version von PGP 2.6 verwendeten RSAREF-Routinen zulassen und auch deutlich mehr, als vom Standpunkt der

◁ Natürlich versucht PGP, alle bekannten Arten schwacher Schlüssel zu vermeiden.

⊤ Das einzige bekannte beweisbar sichere Verfahren ist ein konventioneller Algorithmus namens one-time pad. Leider ist er für die Praxis so gut wie unbrauchbar, da er einen wirklich geheimen Schlüssel voraussetzt, der wirklich zufällig ist und ganz, ganz wirklich nur ein einziges mal verwendet wird. Er wird von Spionen immer wieder verwendet – teilweise aus der Not heraus mehrmals, was die Entschlüsselung ermöglicht. In den USA lief noch bis in die 70er Jahre das Projekt Venona, in dem die so verschlüsselten Nachrichten sowjetischer Spione aus dem zweiten Weltkrieg untersucht wurden.

Geschwindigkeit her sinnvoll ist. Wenn wir annehmen, daß IDEA keine besonderen Schwachstellen enthält, ist der Schwachpunkt, an dem ein kryptanalytischer Angriff ansetzen würde, RSA und nicht IDEA. (Der RSA-Schlüssel ist auch insofern „wertvoller“, als daß er über einen viel längeren Zeitraum verwendet wird als der IDEA-Schlüssel einer einzelnen Nachricht.) Außerdem kann es bei der Verwendung eines reinen RSA-Algorithmus zusätzliche Konstellationen geben, die Ansatzpunkte für einen Angriff ergeben. Fachleute schätzen, daß ElGamal und DSS-Signaturen (die „neuen Schlüssel“ ab PGP 5.0) bei gleicher Schlüssellänge noch sicherer sind als RSA.

4.5. Datenkomprimierung

Normalerweise komprimiert PGP den Klartext, bevor er verschlüsselt wird. Verschlüsselte Daten lassen sich gewöhnlich nicht mehr komprimieren. Die Kompression spart Übertragungszeit und Festplattenkapazität, und, viel wichtiger, sie erhöht die Sicherheit der Verschlüsselung. Die meisten kryptanalytischen Techniken gehen von der Redundanz des Klartextes aus, um die Verschlüsselung zu knacken. Die Datenkompression reduziert die Redundanz des Klartextes und erhöht dadurch wesentlich die Sicherheit vor kryptanalytischen Angriffen. Die Datenkompression kostet zwar etwas Rechenzeit, aber vom Standpunkt der Sicherheit aus ist das gerechtfertigt.

Dateien, die für eine Kompression zu klein sind, oder die sich nicht gut komprimieren lassen, werden von PGP unkomprimiert gespeichert. Bei Bedarf läßt sich auch `bzip2`, `PKzip`, `rar` oder ein anderes Datenkomprimierungsprogramm verwenden, um den Klartext vor der Verschlüsselung zu komprimieren. `PKzip` ist ein weit verbreitetes und effizient arbeitendes Kompressionsprogramm von PKWare Inc. für MS-DOS, das als Shareware vertrieben wird. Sie können auch `zip` benutzen, ein `PKzip`-kompatibles Freeware-Programm für Unix und andere Betriebssysteme von JEAN-LOUP GAILLY. Natürlich gibt es auch `PKzip`-kompatible Programme mit Windows-Oberflächen, beispielsweise `WinZip`. Die Verwendung von `PKzip/zip, tar` oder `rar` hat unter Umständen Vorteile gegenüber der internen Kompression PGPs, weil diese Programme im Gegensatz zu PGP in der Lage sind, mehrere Dateien in einer einzigen komprimierten Datei zusammenzufassen. Bei der Dekompression werden natürlich wieder die einzelnen Originaldateien erzeugt. Die Verwen-

dung von `bzip2` macht vor allem dann Sinn, wenn die Datenmengen wirklich groß werden, da die Kompression in der Regel deutlich besser ist als bei den anderen genannten Verfahren. Auf der beiliegenden CD finden Sie im Verzeichnis `Tools` eine Auswahl verschiedener Packprogramme.

PGP versucht nicht, eine bereits komprimiert vorliegende Klartext-Datei erneut zu komprimieren. Die Empfängerin einer so komprimierten Datei muß sie nach der Entschlüsselung dekomprimieren. Wenn der entschlüsselte Klartext eine komprimierte Datei in einem der Standardformate ist, erkennt PGP dies automatisch und weist den Empfänger darauf hin, daß es sich wahrscheinlich um eine derartige Datei handelt. PGP 2.6.3i setzt auch gleich eine entsprechende Dateinamenserweiterung, so daß auch Systeme, die den Typ einer Datei an der Endung erkennen wollen wie z. B. Windows, damit weiterarbeiten können.

Für die technisch interessierten Leserinnen sei noch erwähnt, daß die aktuelle Version von PGP die Freeware-Kompressions-Routinen enthält, die JEAN-LOUP GAILLY, MARK ADLER und RICHARD B. WALES geschrieben haben. Diese ZIP-Routinen verwenden Algorithmen, die funktionsäquivalent zu denjenigen von `PKzip 2.0` von PKWare sind. Sie wurden im wesentlichen deshalb in PGP eingebaut, weil sie als gut portierbarer C-Quellcode patentfrei zur Verfügung stehen, weil sie wirklich gut komprimieren und weil sie schnell sind.

4.6. Textprüfsummen und digitale Unterschriften

Um eine digitale Unterschrift zu erzeugen, verwendet PGP den geheimen Signaturschlüssel, jedoch nicht für die „Codierung“ des gesamten Textes. Das würde zuviel Rechenzeit kosten• Statt dessen verwendet PGP eine „Textprüfsumme“ und codiert diese mit dem geheimen Schlüssel.

Diese Textprüfsumme ist ein kleines, 128 oder 160 Bit langes „Destillat“ einer Nachricht, von der Idee her ähnlich einer Quersumme oder einer CRC-Summe, falls Ihnen das etwas sagt. Man kann sich die Textprüfsumme auch als eine Art Fingerabdruck der Nachricht vorstellen.▽

• Außerdem gibt es eine Reihe von Angriffen auf die Sicherheit diverser Signatursysteme, wenn ein Angreifer Unterschriften unter (fast) beliebige (sinnlose) Nachrichten erhalten kann.

▽ Der „fingerprint“ eines Schlüssels ist im wesentlichen nichts anderes als die Prüfsumme des Schlüssels, wenn dieser als Nachricht aufgefaßt wird.

Die Textprüfsumme „repräsentiert“ die Nachricht in dem Sinne, daß sich bei (praktisch) jeder Änderung an der Nachricht eine andere Textprüfsumme für die Nachricht ergibt. An der Textprüfsumme läßt sich also zuverlässig erkennen, ob sich ein Fälscher an der Nachricht zu schaffen gemacht hat. Die Textprüfsumme wird durch eine kryptographisch zuverlässige „Einweg-Hash-Funktion“ (engl. *to hash* = zerhacken) berechnet. Ein Fälscher kann wegen des immensen erforderlichen Rechenaufwands keine zweite Nachricht produzieren, die dieselbe Textprüfsumme wie die Originalnachricht hat. In dieser Hinsicht ist das bei PGP verwendete Verfahren zur Berechnung der Textprüfsumme wesentlich besser als die üblichen Quersummen oder CRC-Summen, bei denen es einfach ist, zu einer gegebenen Nachricht eine zweite Nachricht zu finden, die die identische Quer- oder CRC-Summe hat. Andererseits kann aus der Textprüfsumme ebensowenig wie aus einer Quer- oder CRC-Summe die Originalnachricht berechnet werden. Die Länge der Prüfsumme in Bits (128 oder 160) ist entscheidend für den Rechenaufwand, den ein Angreifer betreiben müßte, um zwei Nachrichten mit derselben Prüfsumme oder (was sehr viel aufwendiger ist) eine (zweite) Nachricht zu einer gegebenen Prüfsumme zu finden.

Die Textprüfsumme alleine reicht nicht aus, um die Echtheit einer Nachricht zu kontrollieren. Der Algorithmus für ihre Berechnung ist allgemein bekannt, und er verwendet keinen geheimen Schlüssel. Wenn man die Textprüfsumme einfach so zur Nachricht hinzufügte, könnte ein Fälscher die Nachricht ändern und die Prüfsumme für die geänderte Nachricht neu berechnen. Für eine zuverlässige Kontrolle der Echtheit einer Nachricht muß die Absenderin die Prüfsumme mit ihrem geheimen Schlüssel codieren. Erst hierdurch entsteht eine authentische Unterschrift.

Die Textprüfsumme wird von dem PGP-Programm der Absenderin einer Nachricht berechnet. Die digitale Unterschrift entsteht dadurch, daß die Absenderin die Textprüfsumme und die Zeitmarke mit ihrem geheimen Signaturschlüssel codiert. Die Absenderin verschickt Nachricht und digitale Unterschrift an den Empfänger. Dessen PGP-Programm erhält die für die Unterschrift verwendete Textprüfsumme, indem es die digitale Unterschrift mit dem öffentlichen Signaturschlüssel der Absenderin decodiert. Zur Kontrolle wird die Textprüfsumme aus der erhaltenen Nachricht berechnet. Wenn die aus der Nachricht berechnete Textprüfsumme und die in der Unterschrift enthaltene Prüfsumme übereinstimmen, ist gewährleistet, daß die Nachricht nicht geändert wurde und daß

sie von derjenigen Person stammt, deren öffentlicher Schlüssel für die Kontrolle der Unterschrift verwendet wurde. Bei Verwendung von RSA werden hierbei i. A. dieselben geheimen und öffentlichen Schlüssel wie bei der Verschlüsselung von Nachrichten verwendet, bei ElGamal/DSS-Schlüsseln sind die Schlüssel zum Verschlüsseln und die Schlüssel zum Unterschreiben voneinander unabhängig.

Ein Fälscher müßte die Nachricht entweder so ändern, daß die Textprüfsumme gleich bleibt, was praktisch ausgeschlossen ist, oder er müßte mit der geänderten Textprüfsumme eine neue digitale Unterschrift erzeugen, was ohne den geheimen Schlüssel der Absenderin auch praktisch nicht möglich ist.

Eine digitale Unterschrift beweist, wer eine Nachricht abgeschickt hat, und ob die Nachricht geändert wurde, sei es aufgrund eines technischen Fehlers oder absichtlich. Ebenso verhindert sie, daß ein Absender die Unterschrift unter einer Nachricht leugnen kann.

Bei Verwendung von ElGamal/DSS ist die Verwendung zweier voneinander unabhängiger Schlüssel für Unterschriften und Verschlüsselungen verfahrensbedingt, da DSS nicht für die Verschlüsselung geeignet ist und nicht jeder ElGamal-Schlüssel für Unterschriften eingesetzt werden kann. Diese Trennung ist auch bei Verwendung von RSA-Schlüsseln durchaus sinnvoll, insbesondere, da Schlüssel bei jedem Verschlüsselungssystem in regelmäßigen Abständen gewechselt werden sollten. Es ist sinnvoll, den für Verschlüsselung eingesetzten Schlüssel häufiger zu wechseln als den für Unterschriften eingesetzten, da letzterer die Unterschriften anderer Benutzer trägt. Die Version 2.6.3in unterstützt dies explizit, bei anderen Versionen der 2.6-Reihe erfordert es Mitarbeit des Benutzers, so etwas zu tun. Im neuen Datenformat (PGP 5.x/6.x, OpenPGP) ist es auch möglich, eine ganze Hierarchie von „Teilschlüsseln“ mit unterschiedlicher Bedeutung (angegeben durch Kommentare und Benutzungshinweise wie „nur zur Verschlüsselung“) und Gültigkeitsdauern als einen Schlüssel anzusehen – dann sollte einer der Schlüssel nur dafür verwendet werden, die anderen Schlüssel zu signieren, und dieser Schlüssel sollte dann die Unterschriften von anderen Personen bekommen.

Die Verwendung der Textprüfsumme für die digitale Unterschrift hat neben der Geschwindigkeit noch weitere Vorteile im Vergleich zur Codierung der gesamten Nachricht mit dem geheimen Schlüssel. Die Unterschriften haben alle die gleiche geringe Länge, unabhängig von der Größe der jeweiligen Nachricht. Sie ermöglichen der Software eine au-

tomatische Kontrolle der Korrektheit einer Nachricht, ähnlich wie Quer- oder CRC-Summen. Die Unterschrift kann getrennt von der Nachricht gespeichert werden, bei Bedarf sogar in einem öffentlich zugänglichen Archiv, ohne daß vertrauliche Informationen aus der Nachricht offengelegt werden, weil es unmöglich ist, aus der Kenntnis der Textprüfsumme irgend etwas über den Inhalt der Nachricht zu ermitteln. Darauf basierend, können Sie die Prüfsumme oder auch eine abgetrennte PGP-Unterschrift veröffentlichen, ohne die betreffende Nachricht selbst aus der Hand geben zu müssen. Das kann beispielsweise für Vorhersagen Sinn machen, für Nachrichten oder Pressemitteilungen, aber auch bei Nachrichten, die nie veröffentlicht werden sollen, wie Verträgen, kann es beruhigend sein, zu wissen, daß an einer öffentlichen Stelle genug über den Text dokumentiert ist, damit er nicht manipuliert werden kann. Eine gute öffentliche Stelle für so etwas ist LUTZ DONNERHACKES „ewiges Logfile“ unter `http://www.iks-jena.de/mitarb/lutz/logfile/`.

Die bei PGP verwendeten Algorithmen für die Berechnung der Textprüfsumme sind MD5 (Message Digest 5), von der RSA Data Security Inc. für Public Domain Verwendung freigegeben, sowie SHA1 aus dem „digital signature standard“ (DSS). Inzwischen sind Angriffe auf MD5 gefunden und publiziert worden, die MD5 zwar noch nicht brechen können, es aber ratsam erscheinen lassen, von diesem Algorithmus wegzumigrieren. Er wird noch unterstützt, um Kompatibilität zu älteren PGP-Implementierungen zu bieten.

5. Angriffsmöglichkeiten

Kein Datensicherheitssystem ist unangreifbar. Ein Sicherheitsexperte brachte es einmal genau auf den Punkt, indem er sagte: „Ziel von Sicherheitsmaßnahmen kann es immer nur sein, das Gleichgewicht zuungunsten des Angreifers zu verschieben.“ Auch die Sicherheit von PGP kann auf vielerlei Art ausgehebelt werden. Bei jedem Datensicherheitssystem müssen die Anwender beurteilen, ob die Daten, die geschützt werden sollen, für den Angreifer so viel Wert haben, daß sich für ihn der Aufwand eines Angriffs lohnt. Dies kann durchaus zu der Entscheidung führen, sich nur vor simplen Angriffen zu schützen, ohne sich um aufwendige Angriffe Gedanken zu machen.

5.1. Bekannt gewordenes Mantra und bekannt gewordener geheimer Schlüssel

Sie sollten das Mantra Ihres geheimen Schlüssels auf gar keinen Fall irgendwo aufschreiben oder ein leicht zu ratendes Mantra wählen. Falls jemand dieses Mantra lesen kann und ihm dann noch die Datei mit dem geheimen Schlüssel in die Hände fällt, kann er alles tun, was die eigentliche Schlüsselinhaberin damit auch tun kann, insbesondere alle verschlüsselten Nachrichten lesen und mit dem geheimen Schlüssel gefälschte digitale Unterschriften erzeugen.

Offensichtliche Paßworte, die einfach zu raten sind – wie beispielsweise Namen, insbesondere von Kindern oder der Partnerin – sind ungeeignet. Ein einzelnes Wort (auch einer Fremdsprache) als Mantra kann ebenfalls leicht geraten werden, indem ein Computer die Wörter eines Lexikons solange als Paßwörter ausprobiert, bis das richtige gefunden ist. (Hierfür existieren fertige Programme, die auch leichte Variationen wie rückwärts schreiben oder ‚o‘ durch ‚0‘ ersetzen ausprobieren.) Deshalb ist eine Kombination mehrerer Wörter, von uns „Mantra“ genannt, wesentlich besser als ein einfaches Paßwort. Ein verfeinerter Angriff könnte darin bestehen, einen Computer ein Lexikon

mit berühmten Zitaten durcharbeiten zu lassen, um das Mantra zu finden. Ein leicht zu merkendes, aber schwer erratbares Mantra läßt sich bequem aus ein paar kreativ sinnlosen Sprüchen oder weithin unbekannten literarischen Zitaten zusammenstellen. Beispiele hierfür wären „Earl grey, lauwarm.“ oder „Und Jimmi ging zum Regenbogen.“♠ Wenn Sie schnell und präzise tippen können, ist ein längerer Text natürlich sicherer.

5.2. Fälschung öffentlicher Schlüssel

Eine der gefährlichsten Angriffsmöglichkeiten besteht darin, daß ein öffentlicher Schlüssel gefälscht werden kann. Dies ist *der* wirklich bedeutende und ernsthafte Ansatzpunkt für das Knacken bzw. Umgehen eines Systems, das mit öffentlichen Schlüsseln arbeitet, unter anderem deswegen, weil die meisten Neulinge die Gefahr nicht sofort erkennen.

Damit ist dem Mißbrauch Tür und Tor geöffnet: Beispielsweise ist es für einen Systemadministrator damit ein leichtes, den Mailverkehr seiner Ex-Freundin zu überwachen und zu manipulieren.△ Nähere Einzelheiten und geeignete Gegenmaßnahmen sind detailliert im Abschnitt 7.1 auf Seite 58 beschrieben.

Zusammengefaßt: Wenn Sie einen öffentlichen Schlüssel für die Verschlüsselung einer Nachricht oder für die Prüfung einer Unterschrift verwenden wollen, muß sichergestellt sein, daß er nicht gefälscht ist. Der Echtheit eines neu erhaltenen öffentlichen Schlüssels sollte man nur dann vertrauen, wenn man ihn unmittelbar, auf sicherem Weg, von seinem Besitzer erhalten hat, oder wenn seine Echtheit von jemandem bestätigt ist, dem man vertraut. PGP unterstützt Sie dabei, näheres finden Sie in Kapitel 7.3.

Man muß auch dafür sorgen, daß kein Fremder Änderungen an dem eigenen öffentlichen Schlüsselbund vornehmen kann. Man braucht

♠ Eine amüsante, aber definitiv wahre Begebenheit am Rande: Nachdem ich dieses willkürlich gewählte Beispiel in den Text genommen hatte, bekam ich einen Anruf eines Korrekturlesers mit der entsetzten Frage: „Woher kennst Du mein Mantra?“ Einige Zeit später erfuhr ich, daß es sich um den Titel eines Buches von MARIO SIMMEL handelt – in dem es u. a. um diesen Satz als Paßwort geht…

△ Falls Ihnen das weit hergeholt erscheint, müssen wir Sie leider enttäuschen: In den meisten größeren Firmen wissen die Systemadministratoren weit mehr über die Firmen- und Privatgeheimnisse, als sie wissen sollten – zum großen Teil aus (unverschlüsselten) E-Mails.

physikalische Kontrolle sowohl über den Bund mit öffentlichen Schlüsseln wie auch über den Bund mit geheimen Schlüsseln. Am besten aufgehoben sind diese beiden Dateien auf dem eigenen PC, um einiges schlechter auf einem am Ende auch noch räumlich weit entfernten Mehrbenutzer-Rechner. Auf jeden Fall sollte man Sicherheitskopien der beiden Schlüsseldateien haben (siehe auch den Abschnitt `BAKRING` in Kapitel 14 auf Seite 106).

5.3. Schutz gegen gefälschte Zeitangaben

Eine nicht ganz leicht verständliche Angreifbarkeit von PGP betrifft unehrliche Benutzer, die gefälschte Zeitangaben für die Bestätigung ihrer öffentlichen Schlüssel und ihre Unterschriften verwenden.

Nichts hindert eine unehrliche Benutzerin daran, die Einstellung von Datum und Zeit auf ihrem Computer zu ändern und bei dieser falschen Datumseinstellung ihre Schlüsselbestätigungen und Unterschriften zu erzeugen. So kann diese unehrliche Benutzerin es so erscheinen lassen, als habe sie eine Unterschrift viel früher oder später geleistet, als es tatsächlich der Fall ist, oder als habe sie ihr Paar von öffentlichem und geheimem Schlüssel zu einem anderen Zeitpunkt generiert. Dies kann von juristischem oder finanziellem Nutzen sein. Beispielsweise kann dadurch ein Schlupfloch entstehen, um eine Unterschrift nicht anerkennen zu müssen.

Abhilfe bieten könnten allgemein vertrauenswürdige Institutionen oder Notare, die notariell beglaubigte Unterschriften mit einer vertrauenswürdigen Zeitangabe machen können. Derartige Zeitstempeldienste finden Sie u. a. unter `http://www.itconsult.co.uk/stamper.htm`. Nach dem neuen Signaturgesetz können derartige Zeitstempeldienste durchaus auch in Deutschland rechtskräftige Bestätigungen ausstellen – allerdings nicht direkt mit PGP. Näheres zum SigG finden Sie in Abschnitt 6.4.

Wenn Sie auf die Verwendung der Techniken nach SigG verzichten können, setzt die Idee der Zeitstempeldienste nicht notwendigerweise eine zentrale Institution voraus. Unter Umständen kann jeder vertrauenswürdige Vermittler oder eine unbeteiligte dritte Person diese Aufgabe wahrnehmen, ähnlich öffentlich bestellten Notaren. (Natürlich kann diese Funktion auch von einem echten Notar übernommen werden.) Die Bestätigung eines öffentlichen Schlüssels könnte von dem „Notar“ unterschrieben werden. Der „Notar“ könnte über solche Bestätigungen

Protokoll führen. Das Protokoll wäre öffentlich einsehbar. Rechtskraft könnte eine derartige Unterschrift durch eine gegenseitige Vereinbarung zweier Personen erlangen, ihre Signaturen gegenseitig anzuerkennen.

5.4. Nicht richtig gelöschte Dateien

Ein weiteres potentielles Sicherheitsproblem entsteht durch die Art und Weise, wie bei den meisten Betriebssystemen Dateien gelöscht werden. Wenn man eine Klartext-Datei verschlüsselt und danach löscht, löscht das Betriebssystem die Daten nicht physikalisch. Es markiert nur diejenigen Datenblöcke der Festplatte oder Diskette als „gelöscht“, die den Inhalt der „gelöschten“ Datei enthalten, so daß sie für die Speicherung anderer Daten freigegeben werden. Das ist das gleiche, als würde man vertrauliche Papiere einfach zum Altpapier legen, anstatt sie von einem Reißwolf kleinhäckseln zu lassen. Die Blöcke auf der Festplatte enthalten nach wie vor die originalen vertraulichen Daten und werden vielleicht in naher oder ferner Zukunft durch neue Daten überschrieben. Wenn ein Angreifer diese „gelöschten“ Datenblöcke kurz nach ihrer Freigabe liest, hat er einige Aussicht, den kompletten Klartext zu erhalten.

Genau dies, die Wiederherstellung einer schon vor langem gelöschten Datei, kann sogar ganz unabsichtlich passieren, wenn aus irgend einem Grund etwas mit der Festplatte schief geht und wichtige Dateien gelöscht oder beschädigt sind. Die übliche Rettungsmaßnahme besteht darin, ein Dateiwiederherstellungsprogramm die Dateien reparieren zu lassen. Hierbei werden häufig auch alte, schon vor dem Unfall gelöschte Dateien wieder ausgegraben. Auf diese Art können auch vertrauliche Daten wieder ans Tageslicht kommen, von denen man annahm, daß sie für immer gelöscht seien. Folglich können diese Daten von jedem gelesen werden, der Zugriff auf die betrffende Diskette oder Festplatte hat.

Darüber hinaus legen die meisten Textverarbeitungen Sicherungskopien an und erzeugen häufig auch aus technischen Gründen eine oder mehrere temporäre Dateien, die den gesamten Text oder Teile davon enthalten. Diese Dateien werden vom Textverarbeitungsprogramm automatisch gelöscht – aber eben nur in dem Sinne, daß die Blöcke auf der Festplatte oder Diskette für ein Überschreiben freigegeben werden. Der Text selbst steht nach wie vor in diesen Blöcken.

Hierzu eine wahre Begebenheit: Eine Freundin von PHIL ZIMMERMANN, verheiratet und Mutter kleiner Kinder, hatte eine kurze und nicht

sehr ernstzunehmende Liebesaffäre. Sie schrieb ihrem Liebhaber auf dem Computer einen Brief, und nachdem sie den Brief abschickte, löschte sie die Datei. Nachdem die Affäre schon vorbei war, ging die Diskette kaputt, auf der der Brief gespeichert war. Weil die Diskette einige andere wichtige Daten enthielt, bat sie ihren Ehemann, die Diskette zu reparieren. Der ließ die Diskette von einem Datenwiederherstellungsprogramm bearbeiten, wobei neben den von seiner Frau benötigten Dateien auch der besagte Brief wieder zu Tage kam, was eine Kette tragischer Ereignisse auslöste.

Um zu verhindern, daß der Klartext irgendwann einmal ausgegraben wird, gibt es nur den einen Weg, die „gelöschten“ Daten auch wirklich zu überschreiben. Solange man nicht wirklich sicher weiß, daß die freigegebenen Blöcke der Festplatte oder Diskette sehr schnell wieder mit anderen Daten überschrieben werden, muß man selbst dafür sorgen, daß die Klartext-Datei und die temporären Dateien, die das Textverarbeitungsprogramm angelegt hat, wirklich überschrieben werden. Die PGP-Versionen 2.6.x kennen die Option `-w`, bei deren Verwendung die Klartextdatei mit pseudozufälligen Werten überschrieben wird. PGP 5.5.3i und PGP 6.0i haben ebenfalls eine entsprechende Option. Wenn Ihr Betriebssystem eine Datei beim Überschreiben nicht an einen anderen Ort der Festplatte legt,⊖ sind Sie damit gegen das Wiederherstellen der Dateien durch „OTTO NORMALVERBRAUCHER“ geschützt. Wenn Ihr Betriebssystem Dateien wandern läßt oder Sie sich Gedanken über sonstige temporäre Dateien machen, sollten Sie ein passendes Utility verwenden, das sämtliche Textfragmente, ob aus der „richtigen“ Klartext-Datei oder aus temporären Dateien, löscht, indem es alle freien Blöcke einer Festplatte oder Diskette überschreibt. Sie finden derartige Tools auf der beiliegenden CD im Verzeichnis `cfs`.

Eine weitere Stelle, an der bei den meisten Betriebssystemen „Textspuren“ verbleiben können, sind die „swap files“ (auch Auslagerungsdateien genannt) bzw. Swap-Partitionen: Wenn ein Programm mehr Speicher benötigt, als real im Computer vorhanden ist, wird ein Teil des Hauptspeicherinhalts in diese Auslagerungsdatei bzw. -partition „ausgelagert“, so daß der Hauptspeicher gewissermaßen auf die Festplatte „verlängert“ wird. Auch in dieser Auslagerungsdatei können Teile eines Textes stehen. MS-DOS kennt keine Auslagerungsdateien – endlich mal ein Vorteil dieses Betriebssystems, wenn auch nur im Hinblick auf

⊖ Das kann beispielsweise bei komprimierten Datenträgern der Fall sein.

„Spurensicherheit“. Aber schon Microsoft Windows benutzt eine Auslagerungsdatei, und zwar nicht zu knapp. Die Swap-Dateien/Partitionen lassen sich im Allgemeinen nur überschreiben, wenn das System gerade nicht läuft. GnuPG und auch (mit einem Patch) PGP 2.6.2i verhindern das Ausswappen sensitiver Daten auf den meisten Unix-Varianten, dafür müssen sie allerdings „setuid root“ installiert sein.

Selbst wenn man den Klartext auf der Festplatte oder Diskette überschreibt, kann ein technisch gut ausgestatteter Angreifer die Daten wiedergewinnen. Geringe magnetische Spuren der Originaldaten bleiben auch nach einem Überschreiben auf der Festplatte oder Diskette. Diese Spuren können unter Umständen mittels spezieller Hardware gelesen werden. Sollten Sie sich ernsthaft Gedanken darüber machen, ob ein Geheimdienst Ihre Daten noch lesen kann, sei hier nur angemerkt, daß beispielsweise die NSA u. a. US-Patent Nr. US 5 264 794 besitzt, in dem beschrieben ist, wie man ein Rasterelektronenmikroskop umbaut, so daß es statt der Oberfläche die Restmagnetisierung einer Festplatte abtastet – kurz gesagt: Wenn ein Geheimdienst an Ihre Daten nicht herankommt, machen die Leute ihren Job nicht bzw. nicht gut genug. PGP zu verwenden, hilft allerdings, daß die Überwachung aufwendiger wird und Geheimdienste nicht routinemäßig alle Bürger ihres (oder eines anderen) Staates überwachen können.

5.5. Viren und Trojanische Pferde

Eine andere Angriffsmöglichkeit könnte ein speziell entwickelter Virus oder Wurm sein, der PGP oder das Betriebssystem infiziert. Dieser hypothetische Virus könnte so entworfen sein, daß er das Mantra, den geheimen Schlüssel oder den entschlüsselten Klartext „mithört“ und unbemerkt in eine Datei schreibt oder über ein Netzwerk zum Autoren des Virus schickt. Er könnte auch das Verhalten von PGP so ändern, daß Unterschriften nicht richtig geprüft werden. So ein Angriff ist einfacher und billiger als ein kryptanalytischer Angriff.

Schutz hiergegen fällt unter das allgemeine Thema des Schutzes gegen Viren. Es gibt einige relativ brauchbare kommerzielle und freie Anti-Virus-Produkte, und es gibt „Hygienemaßnahmen“, deren Beachtung das Risiko einer Virusinfektion erheblich reduzieren kann. Eine umfassende Abhandlung dieses Themas würde den Rahmen dieses Buches sprengen. PGP selbst hat keinerlei inneren Schutz gegen Viren, es

geht davon aus, daß der Rechner, auf dem es benutzt wird, eine „vertrauenswürdige Umgebung“ ist. Ein Fall ist bereits bekannt geworden, in dem eine Word-Datei, in der Usernamen und Paßworte für gewisse WWW-Seiten zu finden waren, ein Makrovirus enthielt, der nach privaten Schlüsselringen sucht und diese an einen fremden Rechner schickte. Es ist zu hoffen, daß alle entsprechenden Viren schnell erkannt werden und daß eine entsprechende Warnung schnell viele Leute erreicht.

Ein ähnlicher Angriff könnte eine geschickte Imitation von PGP sein, die sich im Wesentlichen wie PGP verhält, aber nicht so arbeitet, wie anzunehmen wäre. Beispielsweise könnte diese Imitation absichtlich dahingehend verstümmelt sein, daß Unterschriften nicht mehr korrekt geprüft werden, so daß gefälschte Schlüssel nicht mehr erkannt werden können.

Eine solche Version von PGP – ein „Trojanisches Pferd“ – kann ein Angreifer verhältnismäßig einfach erstellen, weil der Quellcode von PGP weit verbreitet ist. Hierzu müßte er nur den Quellcode dahingehend manipulieren, daß eine Imitation von PGP entsteht, die zwar echt aussieht, jedoch nur die Anweisungen ihres teuflischen Meisters ausführt. Dieses „Trojanische Pferd“ im PGP-Mantel könnte weit verteilt werden, mit dem Anschein, es käme von PHILIP ZIMMERMANN bzw. PGP Inc. Wie hinterhältig.

Die allgemeine Verfügbarkeit des Quellcodes von PGP hat aber auch einen anderen, vertrauensschaffenden Aspekt: Die entsprechenden Programmierkenntnisse vorausgesetzt, ist es nur noch eine Frage des Fleißes, den Quelltext auf obskure Stellen durchzusehen. Außerdem kann man das Programm neu übersetzen und sich so eine eigene Arbeitsversion erstellen. Der im nächsten Absatz erwähnte Vergleich mehrerer PGP-Versionen aus unterschiedlichen Bezugsquellen sollte aber zumindest für die selbst erstellte Version vorsichtig interpretiert werden: Selbst wenn der gleiche Compiler verwendet wird, besteht immer noch die Möglichkeit, daß die eigene PGP-Version mit der Compiler-Version 12.34a1 übersetzt wird, während das Entwicklerteam Version 12.34b3 verwendet hat. Unterschiede in den PGP-Versionen bedeuten deshalb nicht gleich das Schlimmste. Auf der anderen Seite heißt das aber auch, daß ein neu übersetztes PGP andere Benutzer verunsichern kann. Deshalb sollte man normalerweise besser die Originalversion weitergeben. Bei Versionen ab 5.0 ist es – das sei hier der Deutlichkeit halber noch einmal angemerkt – nicht gestattet, Änderungen am Quelltext oder geänderte Quelltexte oder aus geänderten Quelltexten kompilierte Versionen weiterzugeben.

Man sollte sich die Mühe machen, PGP von einer zuverlässigen Bezugsquelle zu erhalten, was auch immer das heißen mag.♣ Oder man besorgt sich PGP von mehreren unabhängigen Quellen und vergleicht die einzelnen Versionen mit einem geeigneten Programm, z. B. `diff` oder `cmp`, oder auch einer *vertrauenswürdige* PGP-Version, beispielsweise durch Erstellen einer abgetrennten Signatur für eine Datei und Prüfen der Unterschrift mit der anderen Datei – vergessen Sie nur nicht, die Datei mit der Unterschrift anschließend zu löschen, um sie nicht versehentlich zu verbreiten.

Mit Hilfe digitaler Unterschriften ergeben sich weitere Möglichkeiten festzustellen, ob an PGP herumgepfuscht wurde. Wenn eine Person, der man vertraut, eine digitale Unterschrift für die Datei mit dem ausführbaren PGP-Programm leistet und damit garantiert, daß die Datei zum Zeitpunkt der Unterschrift nicht infiziert oder anderweitig verfälscht ist, kann man einigermaßen sicher sein, eine brauchbare Kopie zu haben. Mit Hilfe einer älteren, vertrauenswürdigen Version von PGP kann die Unterschrift für die neue, zunächst zweifelhafte Version kontrolliert werden. Dieser Test setzt natürlich voraus, daß der für die Kontrolle der Unterschrift verwendete öffentliche Schlüssel und seine Eigentümerin einen hohen Grad an Vertrauen hat und daß bereits ein vertrauenswürdiges PGP installiert ist.

5.6. Lücken in der physischen Sicherheit

Einige der bisher besprochenen Sicherheitsprobleme bestehen auch ohne daß ein Angreifer unmittelbaren Zugang zu dem Computer hat, auf dem die geheimzuhaltenden Daten gespeichert sind. Ein direkter Zugriff auf den Computer oder ausgedruckte Texte ist auch denkbar durch Einbruch, Durchsuchen des Mülls, eine unerwartete, evtl. auch unbegründete Hausdurchsuchung, Bestechung, Erpressung oder Bespitzelung. Von einigen dieser Angriffe dürften insbesondere politische Basisorganisationen und ähnliche Gruppierungen betroffen sein, die weitgehend auf ehrenamtliche Mitarbeit angewiesen sind. Die Presse hat einiges darüber berichtet, daß das FBI im Rahmen seines COINTELPRO-Programms mit Einbruch, Infiltration und illegalen Wanzen gegen Antikriegs- und Bürgerrechtsgruppen gearbeitet hat. In Deutschland ist

♣ Wir hoffen, daß die CD zu diesem Buch eine solche Quelle ist. Aber warum sollten Sie uns vertrauen? :-)

es eine bekannte Tatsache, daß der Verfassungsschutz seit den Gründungszeiten die rechts- und linksextremen Netzwerke „Thule“ und „Spinnennetz“ nicht nur überwacht, sondern sogar aktiv mit aufgebaut hat.× Nicht zu vergessen: Die Watergate-Affäre. Natürlich gibt es auch „positive“ Beispiele, wie die allgemeine Telephonüberwachung, mit der deutsche Firmen enttarnt wurden, die Chemiewaffenfabriken nach Lybien verkaufen wollten.

Die Verwendung eines Verschlüsselungsprogramms kann ein trügerisches, einschläferndes Gefühl der Sicherheit entstehen lassen. Kryptographische Techniken schützen Daten aber nur solange, wie sie verschlüsselt sind. Löcher in der unmittelbaren physischen Sicherheit können nach wie vor Klartextdaten und geschriebene oder gesprochene Information offenlegen. Diese Art von Angriffen ist einfacher und billiger als ein kryptanalytischer Angriff auf PGP.

5.7. „Sturmangriffe“ (tempest attacks)

Die folgende Diskussion mag maßlos paranoid erscheinen, aber so eine Einstellung ist für eine fundierte Auseinandersetzung mit möglichen Angriffen durchaus angemessen. Wenn Sie von den in diesem Abschnitt besprochenen Angriffen bedroht zu sein glauben, sollten Sie den Absatz am Ende beachten.

Eine andere Angriffsmöglichkeit für einen gut ausgerüsteten Gegner ist die Auswertung der elektromagnetischen Strahlung, die ein Computer aussendet. Ein solcher Angriff ist zwar teuer und arbeitsintensiv, aber wahrscheinlich immer noch billiger als eine richtige Kryptanalyse. Ein entsprechend ausgerüsteter Kleinbus könnte in der Nähe des abzuhörenden Computers geparkt sein und jeden Tastendruck und jeden Bildschirminhalt aufzeichnen. Das würde alle Paßworte, Nachrichten usw. offenlegen. Abwehren läßt sich dieser Angriff durch eine geeignete Abschirmung des Computers, des Zubehörs (Drucker usw.) und gegebenenfalls der Netzwerk-Verkabelung. Alternativ kann auch der Raum als solcher abgeschirmt werden, in dem der Rechner steht, beispielsweise durch ein geerdetes Kupferdrahtnetz unter der Tapete und dem Teppich. Eine solche Abschirmung ist unter dem Begriff „sturmsicher“ bekannt, und wird von einigen Regierungsbehörden und Rüstungsfirmen ein-

× Das gab ECKEHARD WERTHEBACH, der ehemalige Leiter des Bundesamtes für Verfassungsschutz, jetzt Berliner Innenminister, gegenüber Mitgliedern des FoeBuD e. V. zu.

gesetzt. Es gibt Firmen, die diese Abschirmungen anbieten, allerdings ist der Kauf möglicherweise genehmigungspflichtig. Woher das wohl kommt?

Wer allerdings der Meinung ist, derartig ausfeilten Angriffen ausgesetzt zu sein, sollte sich ohnehin mit einem professionellen Sicherheitsberater in Verbindung setzen.

5.8. Probleme bei Mehrbenutzer-Computern

PGP wurde ursprünglich für (Einplatz-)MS-DOS-Computer entworfen, zu denen nur eine Person (unmittelbaren) Zugang hat. Wenn Sie PGP zu Hause auf Ihrem privaten PC benutzen und solange niemand einbricht oder die elektromagnetischen Signale des PCs auswertet, sind die Klartext-Dateien und die geheimen Schlüssel wahrscheinlich sicher. Sicherheitslücken bei Anschluß an das Internet sind eine ganz andere Frage, insbesondere wenn Sie Back Orifice, DeepThroat, NetSphere, GateCrasher, Portal of Doom, GirlFriend, Hack'a'Tack, EvilFTP, phAse Zero, SubSeven oder dergleichen (das sind Windows-Programme, die anderen Menschen gestatten, Ihren Windows 95/98/NT-Rechner per Internet komplett fernzusteuern, Tasteneingaben mitzulesen, Dateien zu lesen und zu ändern o. ä.) installieren. Wenn ein Angreifer die Sicherheit Ihres Rechners umgehen kann, ist PGP machtlos.

Aber mittlerweile werden zunehmend vernetzte Rechner eingesetzt und die Verbreitung von Multi-User-Systemen wie Unix, Linux oder Windows NT nimmt ebenfalls zu. Einen Rechner mit mehreren Personen zu benutzen, erhöht das Risiko, daß Klartext-Dateien, Schlüssel oder Paßworte von Unbefugten gelesen werden.♢ Der Systemverwalter oder ein versierter Eindringling kann die Klartext-Dateien lesen und unter Umständen auch mittels spezieller Software heimlich die Tastatureingaben und Bildschirmausgaben mitlesen. Auf vielen Unix-Systemen kann jede Benutzerin mit dem Kommando `ps` einiges an Informationen über die Prozesse der anderen Benutzer erhalten, beispielsweise alle Umgebungsvariablen⊗ und die gesamte Aufrufzeile.⊙

♢ Für Rechner, die nur von Ihnen verwendet werden und die kein Netzwerk haben, gelten diese Überlegungen natürlich nicht, unabhängig vom verwendeten Betriebssystem.

⊗ Zum Ausprobieren: `ps aex`

⊙ PGP löscht den in Abschnitt 15.21 besprochenen Parameter `-z` samt Paßwort von der Kommandozeile, aber bis es das tun kann, kann ein anderes Programm Ihr Mantra bereits gelesen haben.

Das tatsächliche Sicherheitsrisiko hängt von der jeweiligen Situation ab. Ein Mehrbenutzer-Rechner kann sicher sein, wenn man allen Benutzern traut und wenn die Sicherheitsmechanismen ausreichen, um Eindringlingen standzuhalten, oder auch, wenn es ganz einfach keine hinreichend interessierten potentiellen Eindringlinge gibt. Manche Systeme sind schon dadurch sicher, daß sie von nur einer Person benutzt werden – es gibt bereits Notebooks und Palmtops, die mit Unix arbeiten. PGP vollkommen von der Benutzung unter Unix auszuschließen, wäre unsinnig. Dasselbe gilt für Windows 95/98/NT.

PGP ist nicht dafür gedacht, Daten zu schützen, die als Klartext auf einem schlecht geschützten oder „aufgeflogenen“ Rechner vorhanden sind. Ebensowenig kann es einen Eindringling davon abhalten, einen geheimen Schlüssel während seiner Benutzung mitzulesen. Diese Risiken muß man sich gerade für Mehrbenutzer-Rechner klarmachen und die Erwartungen an PGP und das eigene Verhalten darauf abstimmen. Aber vielleicht hat die Leserin doch die Möglichkeit, PGP auf einem „isolierten“, also nicht an ein Netzwerk angeschlossenen Ein-Platz-PC zu verwenden, der unter ihrer unmittelbaren physischen Kontrolle ist. Auf diese Weise setzen wir PGP nach Möglichkeit ein, und dazu raten wir auch.

5.9. Statistik von Nachrichtenverbindungen

Selbst wenn ein Angreifer nicht in der Lage ist, den Inhalt der verschlüsselten Nachrichten zu lesen, hat er immer noch die Möglichkeit, brauchbare Informationen daraus zu gewinnen, woher Nachrichten kommen, an wen sie gehen, wie lang sie sind oder wann sie geschrieben wurden. Dies entspricht der Auswertung von Telephonverbindungen, ohne daß die einzelnen Gespräche abgehört werden, beispielsweise aufgrund eines im Altpapier gefundenen Einzelgesprächsnachweises. Das ist mit „Statistik von Nachrichtenverbindungen“ („traffic analysis“) gemeint.

PGP schützt hiervor nicht. Dieses Problem können Sie nur auf der Ebene angehen, auf der Ihre Nachrichten versandt werden, und damit hat PGP zunächst einmal nichts zu tun. Es gibt aber Ansätze, dieses Problem zu lösen, beispielsweise die (meist auf PGP aufbauenden) Remailer oder auch mixmaster, ein deutlich aufwendigeres Verfahren.

Anonyme Remailer

Unter diesem Begriff verstehen wir Programme, die (PGP-verschlüsselte) Nachrichten entgegennehmen, sie entschlüsseln und an eine im verschlüsselten Text angegebene Adresse weiterversenden. Eine Liste derartiger Remailer mit aktuellen Statistiken über Zuverlässigkeit und Geschwindigkeit finden Sie unter `http://drule.org/remailer/` oder auch `http://www.anon.efga.org/`.

Eine Nachricht an einen Remailer vom hier besprochenen Typ (der „cypherpunk type-I“ genannt wird) hat folgenden Aufbau[∨]:

1. Die erste Zeile der Nachricht besteht aus '`::`'.
2. Dann kommen – ohne Leerzeilen – Zeilen im Header-Format; die wichtigste ist `Anon-To: ad@res.se`.
3. Eine Leerzeile
4. Wenn Sie weitere Header in der versandten Mail haben wollen (beispielsweise einen Betreff):
   - Die Zeile '`##`'.
   - Die gewünschten Headerzeilen.
   - Eine Leerzeile.
5. Der Text Ihrer Nachricht.

Ein Beispiel:

```
::
Anon-To: ct@ct.heise.de,ccc@ccc.de,foebud@foebud.org

##
Subject: Bedenkliches internes Memorandum bei XYZ-Software GmbH

Sehr geehrte Damen und Herren,

 anbei ein Memorandum, das mich heute erreichte.  Bitte haben Sie
dafür Verständnis, daß ich meine Arbeitsstelle nicht riskieren
möchte und Ihnen daher keine Antwortadresse nennen kann.
...
```

∨ Sie *können* die Nachrichten auch in anderen Formaten generieren. So, wie hier beschrieben, sollte es aber immer funktionieren.

Diese Nachricht könnten Sie so an einen der Remailer schicken – aber dann wäre der Inhalt für jeden zu lesen. Die Remailer haben daher eigene PGP-Schlüssel, mit denen Sie die Nachrichten verschlüsseln sollten. In diesem Fall empfiehlt es sich (bei manchen Remailern muß es gemacht werden), vor den verschlüsselten Text die Zeilen '`::`' und '`Encrypted: PGP`' zu setzen. Beispielsweise an `remailer@foebud.org` sieht der Anfang einer entsprechend verschlüsselten Nachricht so aus:

```
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3ia

hIwCmjyrXLROh0cBBADYbWUfytdwaaPQsJ/xljWaXAoUPlGJDpFjvxzVgtbARVq2
d+XunemKNxguE29BsFZYZ0LOGAqyWyNpXovtbR1VcrZo1JWsvXW2FXSazvRGLkYI
...
```

Empfehlenswerter wäre es natürlich, mehrere Remailer in dieser Weise hintereinanderzuhängen. Das Vorgehen hierfür ist Folgendes:

1. Verschlüsseln Sie ihre Nachricht an die Empfängerinnen.
2. Fügen Sie die nötigen Zeilen hinzu, um einem Remailer mitzuteilen, an wen die Nachricht geschickt werden soll. (Also eine Zeile '`::`' und eine Zeile `Anon-To: ad@res.se`.)
3. Um einen weiteren Remailer zu verwenden, gehen Sie zurück zu Schritt 1, wobei Sie den gerade gewählten Remailer als Empfänger betrachten.

Glücklicherweise gibt es Programme, die Ihnen dabei helfen – schauen Sie sich am besten auf der mitgelieferten CD im Verzeichnis `remailer` um.

Mixmaster

Deutlich komplexer gestalten sich die sogenannten „mixmaster“. Das ist ein System von Remailern, die eine zu versendende Nachricht anonymisieren, in Blöcke ungefähr konstanter Größe aufteilen, diese Blöcke auf beinahe zufälligen Wegen zu einem End-Remailer senden und von

dort die Nachricht an den Empfänger versenden. Zusätzlich werden durch die mixmaster Datenblöcke erzeugt und versandt, die keinen Inhalt enthalten und irgendwo einfach wieder verschwinden. Eine erfolgreiche Analyse, wer eine Nachricht auf diesem Weg an wen sendet, würde ein Zusammenarbeiten aller Betreiber der beteiligten mixmaster-Programme erfordern, und selbst dann wäre es nicht einfach.

Um Nachrichten mit mixmaster zu versenden, müssen Sie zunächst das Programm selbst installieren. Sie finden die nötigen Dateien auf der beiliegenden CD im Verzeichnis `Remailer`. Unter MS-DOS können Sie Mixmaster im Zusammenspiel mit Private Idaho verwenden. Nach der Installation können Sie einfach `mix` aufrufen und haben dann die Möglichkeit, mit `m` eine Nachricht zu versenden. Die erscheinenden Menüs dürften im Wesentlichen selbsterklärend sein, eine genauere Anleitung würde den Rahmen dieses Handbuchs sprengen.

5.10. Kryptanalyse

Ein kryptanalytischer Angriff könnte von einem Geheimdienst durchgeführt werden, der über ein ausreichendes Arsenal von Mathematikern und Supercomputertechnologie verfügt. Dieser Angreifer könnte beispielsweise einen RSA-Schlüssel unter Verwendung eines bahnbrechenden neuen, geheimgehaltenen Algorithmus für die Primfaktorzerlegung knacken.

Das ist denkbar, aber man sollte nicht vergessen, daß die US-Regierung dem RSA-Algorithmus soweit vertraut, daß mit ihm (nach Aussage RON RIVESTs, einem der Erfinder des RSA-Algorithmus) Kernwaffen gesichert werden. Laut einem Artikel im Spektrum der Wissenschaft wird auch RSA verwendet, um die Nachrichten der automatisierten Überwachungssanlagen für Nuklearversuche, die die USA und die GUS wechelseitig in ihrem Hoheitsgebiet stationiert haben, zu signieren – sicherlich ein Anwendungsgebiet für digitale Signaturen ohne Verschlüsselung, denn die Stationen sollen ja nur die erlaubten Daten senden, diese sollen aber nicht manipuliert werden können. Im zivilen Bereich gibt es seit 1978 intensive, aber bislang erfolglose, Versuche, RSA zu knacken.

Möglicherweise hat eine Regierung auch geheimgehaltene Methoden, mit denen IDEA oder ein anderer der bei PGP verwendeten konventionellen Verschlüsselungsalgorithmen geknackt werden kann. Das

wäre der zweitschlimmste Alptraum eines jeden Kryptographen.⊕ In der praktischen Kryptographie gibt es keine Garantie für absolute Sicherheit.

Doch nach wie vor ist etwas Optimismus gerechtfertigt. Die bei PGP eingesetzten Algorithmen gehören zu den am besten analysierten Verfahren in der öffentlich zugänglichen Literatur und halten bislang allen bekannten Angriffen stand.

Aber selbst wenn einer der Algorithmen die eine oder andere subtile Schwachstelle haben sollte, komprimiert PGP den Klartext vor der Verschlüsselung, was die von einer solchen Schwachstelle ausgehende Gefährdung um einiges reduzieren dürfte – komprimierte Daten haben weniger Struktur, was einen Angriff schwieriger macht. Der für das Knacken erforderliche Rechenaufwand dürfte in den meisten Fällen um einiges höher sein als der Wert der entschlüsselten Nachricht.

Wenn man in einer Situation ist, in der die Furcht vor so einem Angriff größten Kalibers berechtigt ist, wäre es an der Zeit, die Dienste einer Sicherheitsberaterin in Anspruch zu nehmen, die auf die jeweilige Situation zugeschnittene Lösungen anbieten kann.

Kurz gesagt, ein Gegner kann mühelos, sogar routinemäßig Datenkommunikation abhören, es sei denn, die Daten sind gut kryptographisch geschützt. Wenn man PGP verwendet und die erforderlichen Vorsichtsmaßnahmen beachtet, muß ein Angreifer erheblich mehr Arbeit und Kosten aufbringen, um in die Privatsphäre einzubrechen.

Wenn man sich vor einfachen Angriffen schützt und annehmen kann, daß man nicht einem entschlossenen und sehr gut ausgestatteten Angreifer gegenübersteht, dann dürfte die Verwendung von PGP sicher sein. PGP sorgt für eine prima geschützte Privatsphäre.

⊕ Der schlimmste ist P=NP, aber was das bedeutet, würde hier zu weit führen – wenden Sie sich vertrauensvoll an eine Informatikerin.

6. Politik

Verschlüsselung ist ein hochgradig politisches Thema. Die Frage, ob und mit welchen Mitteln der einzelne Bürger, die private Firma, verschlüsseln darf oder inwieweit staatliche Stellen ein Recht haben (sollen), alles abzuhören, ist ein sehr heißes Eisen.

PGP ist in mehrfacher Hinsicht hochpolitisch, weil es nicht nur ein einfach zu bedienendes Programm mit sehr guter Verschlüsselung ist, sondern darüber hinaus auch noch (bis auf ein paar Versionen) gratis verteilt wird und sich somit als Verschlüsselungssystem für die Massen eignet und etabliert hat. Unsere persönliche Meinung zu diesem Thema ist eine sehr freiheitliche, die im wesentlichen auch den Inhalt des Volkszählungsurteils („informationelle Selbstbestimmung“) widerspiegelt. PGP verschlüsselt private Nachrichten auf Kommunikationswegen, die von fast jedem Interessenten mit minimalem Aufwand abgehört werden können. Ermittlungsbehörden, die einen ernsthaften Verdacht haben, haben auch andere Wege, zu ermitteln. Zu Angriffen auf die elektronischen Nachrichten selbst sei hier auf die Diskussion im Kapitel 5 verwiesen. In der Argumentation gegen private Verschlüsselung werden oftmals unhaltbare Argumente gebracht, beispielsweise sei es nicht möglich, Kinderpornohändler dingfest zu machen, wenn diese Kryptographie einsetzen dürften. Dabei wird beispielsweise übersehen, das diese Kriminellen eine Möglichkeit brauchen, neue Kunden zu finden und von ihnen Geld zu erhalten; selbstverständlich ist hier primär klassische Polizeiarbeit gefragt.

6.1. Warum eigentlich PGP benutzen?

PGP schützt die Privatsphäre. Ob Sie nun eine politische Kampagne planen, über Ihr Einkommen reden oder eine Affäre geheimhalten wollen, ob Sie über etwas reden wollen, das (Ihrer Meinung nach zu Unrecht) illegal ist oder ob Sie Daten speichern, transportieren und versenden müssen, die unter das Datenschutzgesetz fallen (z. B. Systembetreu-

er, die ihre Userdaten über eine Telephonleitung transportieren), ob Sie manchmal Nachrichten schreiben wollen, von denen andere genau wissen sollen, daß sie von Ihnen stammen oder ob Sie eben dies bei Nachrichten von anderen prüfen wollen (z. B. bei elektronischen Bestellungen oder von Ihnen geschriebenen Programmen oder Pressemitteilungen) oder ob Sie einfach nur selbst entscheiden wollen, wer Ihre *private* Post liest – die meisten Menschen, die E-Mail nutzen, werden PGP früher oder später verwenden können.

Wenn Sie sich davor sträuben, Ihre privaten Mails zu verschlüsseln: Warum verwenden Sie eigentlich (verschlossene) Briefumschläge? Nehmen wir einmal an, es sei die gängige Ansicht, brave Bürger bräuchten keine Briefumschläge zu verwenden. Wenn nun irgend jemand aus irgendeinem Grund einen Briefumschlag verwenden würde (mehrere Blätter, ein Liebesbrief, den die Mutter des Adressaten nicht lesen soll etc.), dann wäre dies höchst verdächtig. Glücklicherweise verwenden die meisten Menschen Briefumschläge, doch bei elektronischen Briefen ist dies bislang noch nicht der Fall. Dabei sind die elektronischen Datenwege (rein technisch) sehr viel leichter zu überwachen als konventionelle Briefpost, und elektronische Nachrichten können auch sehr schnell auf bestimmte Reizworte durchsucht werden.

Gehen Sie eigentlich regelmäßig zum AIDS-Test? Möchten Sie sich regelmäßig auf illegalen Drogenkonsum untersuchen lassen? Verlangen Sie nicht einen richterlichen Durchsuchungsbefehl, wenn die Polizei bei Ihnen eine Hausdurchsuchung machen will? Haben Sie am Ende etwas zu verbergen? Wahrscheinlich sind Sie ein Drogendealer, ein Falschparker oder ein Subversiver, wenn Sie Briefumschläge benutzen.

Was wäre, wenn es der allgemeinen Auffassung entspräche, rechtschaffene Bürger sollten all ihre Post auf Postkarten schreiben? Wenn ein braver Mensch auf die Idee käme, sein Briefgeheimnis durch einen Umschlag zu schützen, wäre das höchst verdächtig. Sicherheitsbehörden würden vielleicht jeden Briefumschlag untersuchen, um zu kontrollieren, was er verbirgt. Glücklicherweise leben wir nicht in so einer Welt – die meisten Menschen verwenden Briefumschläge, so daß ein Briefumschlag auch nichts Verdächtiges ist. Es wäre schön, wenn alle E-Mail verschlüsselt würde, ob sie nun verbotene Nachrichten enthält oder nicht, so daß die Verschlüsselung von E-Mail genauso wenig verdächtig wird wie das Verwenden von Briefumschlägen.

Wenn Sicherheitsbehörden das Brief- oder Telephongeheimnis brechen wollen, müssen sie einigen Aufwand treiben. Sie müssen den

Umschlag aus dem Postweg herausfischen, ihn durchleuchten, aus dem Ergebnis mit Hilfe eines aufwendigen Computerprogramms die einzelnen Seiten extrahieren und das Ganze dann lesen.∞ Das Abhören von Telephongesprächen ist sehr zeitintensiv, und auch eine Transskription kostet Zeit. Eine so arbeitsintensive Überwachung kann nicht im großen Stil betrieben werden, von Reizwort-erkennenden Maschinen einmal abgesehen. Aber auch die sind sehr aufwendig und automatisieren lediglich die Vorauswahl derjenigen Texte, die anschließend von Menschen kontrolliert werden.

Ein immer größerer Teil unserer Privatkommunikation läuft über E-Mail. E-Mail aber läßt sich sehr leicht überwachen. Die Suche nach verdächtigen Schlüsselworten ist kein Problem. Das kann ganz einfach routinemäßig und vollautomatisch in großem Maßstab durchgeführt werden, ohne daß es irgendwie auffällt. Internationale Kommunikationswege werden nicht nur in den USA, sondern auch hier in Europa (größtenteils von den britischen Inseln aus) von der National Security Agency (NSA) und anderen Diensten abgehört. Denken Sie auch bei Verschlüsselung nicht nur an den „bösen Staat“, vor dem Sie sich schützen müssen, sondern vor allem auch an die lukrative Wirtschaftsspionage. Darüber hinaus sind es nicht nur Sie selbst, die Sie mit Verschlüsselung schützen können und sollten – denken Sie auch an Ihre Familie und andere Menschen, die Ihnen nahestehen. Nebenbei bemerkt sind auch einige Fälle bekannt geworden, wo staatliche Geheimdienste Wirtschaftsspionage betrieben haben – der französische gibt das sogar offen zu. In letzter Zeit ist es allgemein bekannt geworden, daß US-Regierungsbehörden – vor allem in Zusammenarbeit mit der britischen Regierung – alle europäischen Auslandstelephonate und vermutlich einen Großteil der inländischen Telephonate abhören. Dieses „Echelon“ genannte Programm ist seit vielen Jahren (mindestens seit 1970) in Betrieb, und laut einem kürzlich im Salt Lake Tribune erschienenen Artikel wurden damit auch Amnesty International und Prinzessin DIANA belauscht – nicht unbedingt die ersten Ziele, die uns im Zusammenhang mit nationaler Sicherheit einfallen. Derselbe Artikel behauptet wörtlich: „Firmengeheimnisse europäischer Gesellschaften zu stehlen, ist eine ständig angebotene Dienstleistung.“ Die deutsche Regierung hat sich über lange Zeit geweigert, irgendwelche Kommentare über die Exi-

∞ Die Zeiten, wo Briefe geöffnet werden mußten, um sie zu lesen, dürfte inzwischen weltweit vorbei sein, zumindest, was Sicherheitsbehörden angeht.

stenz oder Bedeutung von Echelon abzugeben. Nähere Informationen zu Echelon und verwandten Themen finden Sie im Magazin Telepolis unter `http://www.heise.de/tp/`. Wir empfehlen ganz allgemein besonders die Artikel von CHRISTIANE SCHULZKI-HADDOUTI.

Wir werden bald den voll ausgebauten „Information Superhighway“ haben, der die Welt kreuz und quer mit Glasfaserkabeln überzieht und die allgegenwärtigen Computer verbindet – manche Leute meinen, wir hätten ihn schon. E-Mail wird zum allgemeinen Standard werden, mehr noch als jetzt. Regierungsbehörden werden für die Verschlüsselung unserer Nachrichten ihre eigene Technologie empfehlen. Viele Leute mögen dieser Technologie vertrauen. Aber andere werden es vorziehen, ihre eigene Wahl zu treffen.

In Staaten „westeuropäischer Bauart“ (also z. B. Deutschland, England, Frankreich, USA, Japan, Schweiz) gibt es innerhalb des Staats- und Regierungsapparates zu so gut wie allen Fragen widerstreitende Faktionen, die versuchen, ihre jeweiligen Ansichten durchzusetzen – mit gutem Recht, denn eigentlich ist genau das ihr Auftrag. Zum Teil gehen diese Entwicklungen leider sehr stark in eine Richtung, die den einzelnen Bürger zu bevormunden und zu überwachen versucht. Beispielsweise gab es 1991 eine Gesetzesvorlage im US-Senat, in der folgendes zu lesen stand: „Der Senat ist der Ansicht, daß die Anbieter elektronischer Kommunikation und die Hersteller elektronischer Kommunikationsgeräte sicherstellen müssen, daß die Regierung Zugriff auf die entschlüsselte Sprachübertragung, Daten- und andere Kommunikation hat, sofern hierfür eine Gesetzesgrundlage vorhanden ist.“ Diese Vorlage wurde nach heftigen Protesten von Bürgerrechtsgruppen und Industrieverbänden zurückgezogen. 1992 gab es einen Vorstoß des FBI, der im wesentlichen dasselbe Ziel zum Inhalt hatte, der Vorschlag wurde abgelehnt, aber 1993 erneut vorgelegt. Die aktuelle Situation in Deutschland läßt sich unter anderem auf den Webseiten CHRISTIANE SCHULZKI-HADDOUTIS nachlesen, `http://members.aol.com/InfoWelt/bestof.html`.

Ebenfalls 1993 wurde ein Gesetzesentwurf diskutiert, nach dem im Wesentlichen nur noch eine bestimmte Verschlüsselungsmethode namens „Clipper“ eingesetzt werden sollte, die von der NSA entwickelt wurde und deren Design geheimgehalten wurde. Der Haken bei Clipper: Bei der Herstellung bekommt jeder Chip seinen individuellen Schlüssel, und die US-Regierung erhält Kopien dieser Schlüssel, um abhören zu können. Aber keine Sorge – die Regierung verspricht, daß diese „Zweitschlüssel“ nur ordnungsgemäß entsprechend den rechtlichen Bestimmungen eingesetzt werden …

Damit Clipper für die Behörden den vollen Nutzen entfalten kann, wäre der nächste logische Schritt ein Verbot anderer Formen von Kryptographie gewesen. Das war nach offiziellen Statements nicht geplant, die US-Regierung hat aber in der Vergangenheit schon einige Empfehlungen dadurch de facto zu Vorschriften werden lassen, daß sie nur noch mit solchen Firmen Geschäfte tätigt, die sie befolgen. Hierbei sollte man nicht vergessen, daß die Hauptaufgabe der NSA, soweit öffentlich bekannt ist, darin liegt, andere Staaten, internationale und US-interne Nachrichtenwege abzuhören, um die Staatssicherheit zu gewährleisten. Nebenbei bemerkt ist das Design des Clipper-Algorithmus inzwischen veröffentlicht und geknackt worden.

Die aktuelle Entwicklung in Deutschland sieht nicht viel rosiger aus: Nach IuKDG (Gesetz über Informations- und Kommunikationsdienstleistungen) sind Diensteanbieter⋆ verpflichtet, den „Bedarfsträgern“, also den staatlichen Ermittlungsbehörden, *auf eigene Kosten* einen Zugang zur Verfügung zu stellen (das läuft auf eine oder mehrere Standleitungen quer durch Deutschland hinaus), über die die Beamten ohne Wissen des Anbieters auf die Kundendaten zugreifen und jederzeit eine Überwachung des Telephon- und E-Mail-Verkehrs veranlassen können. Nimmt man noch hinzu, daß im Jahre 1997 der „große Lauschangriff“ verabschiedet wurde, ein Gesetz, daß das Abhören von Privatwohnungen bis auf wenige Ausnahmen fast zur Routineangelegenheit werden läßt, muß die Frage gestattet sein, weshalb der Staat berechtigt sein soll, derart tiefgreifend in die Privatsphäre unbescholtener Bürger einzugreifen. Diese Frage ist umso bedeutender, wenn man bedenkt, daß die Realität nicht so aussieht, als würden die gewonnenen Erkenntnisse und Daten lediglich zu den Ermittlungszwecken eingesetzt, zu denen sie erhoben wurden. Näheres erfahren Sie in den Tätigkeitsberichten der Datenschutzbeauftragten, die Ihnen die Landesbeauftragten für den Datenschutz auf Anfrage gerne zusenden.

Wenn Privatsphäre kriminell wird, werden nur Kriminelle Privatsphäre haben. Geheimdienste, Drogenkartelle, große Wirtschaftsunternehmen, Militär, sie alle haben gute Kryptographiesysteme. Nur normale Menschen und politische Basisorganisationen sind davon ausgenommen – waren davon ausgenommen, bis es PGP gab. Das ist der Grund, warum PGP geschrieben wurde, und das ist auch der Grund, warum Sie PGP nutzen sollten.

⋆ Zu den Diensteanbietern nach IuKDG gehören auf jeden Fall die Telephongesellschaften einschließlich der Mobilnetzbetreiber, streng nach Wortlaut aber auch alle Internetprovider und MailBox-Systeme.

6.2. Zur rechtlichen Situation der Verschlüsselung in Deutschland

Wir sind selbst keine Juristen und haben auch noch von keinem Juristen ein umfassendes Statement zum Thema Verschlüsselung bekommen können. Entwicklung und Einsatz von Kryptographie sind in Deutschland explizit keinen Beschränkungen unterworfen (außer für Telekommunikationsdiensteanbieter, die, wie oben erwähnt, Zugriff auf die nicht von ihnen verschlüsselten Daten gewähren müssen). Dies hat die Bundesregierung zuletzt am 2. Juni 1999 entschieden. Der Export starker Verschlüsselung unterliegt für „wenige Länder“ Einschränkungen, im Zweifelsfall sollten Sie versuchen, eine Genehmigung des Bundesausfuhramtes (Tel. 06196-90 80) einzuholen.

Im Juli 1996 haben die Länder Argentinien, Australien, Belgien, Bulgarien, Deutschland, Dänemark, Finnland, Frankreich, Griechenland, Irland, Italien, Japan, Kanada, Luxemburg, Neuseeland, die Niederlande, Norwegen, Polen, Portugal, die Republik von Korea, Rumänien, Schweden, die Schweiz, die Slovakei, Spanien, die Türkei, UK (Großbritannien), die USA, die Ukraine, Ungarn, die russische Föderation, die tschechische Republik und Österreich gemeinsam das Abkommen von Wassenaar verfaßt und unterzeichnet, in dem sie sich auf eine gemeinsame Grundlinie verständigen, was den Handel mit Kriegswaffen und „dual-use goods“, also Gütern mit potentiell militärischer Verwendungsmöglichkeit, betrifft. Diese Regelungen betreffen auch Kryptographie, deswegen möchten wir Ihnen einige Eckpunkte erläutern.

Das (offizielle) Ziel des Abkommens ist eine größere Transparenz, Austausch von Ansichten und Informationen sowie ein stärkeres Verantwortungsgefühl beim Handel mit den genannten Gütern. Das übergeordnete Ziel, so der Text, ist die Stabilisierung der politischen Lage weltweit. Die Staaten sind frei in ihren Entscheidungen, es handelt sich beim Abkommen mehr um ein „gentlemen's agreement“ ohne feste Bedeutung; es ist aber anzunehmen, daß die reale Situation sich den Zielen dieses Papieres angleicht.

Das Abkommen umfaßt auch kryptographische Software, die in Bereich 5, Abschnitt 2 fällt. Die dort genannten Bedingungen für Ausnahmen werden von der kommerziellen Version von PGP 5.x/6.x nicht erfüllt, da es einen symmetrischen Algorithmus mit einer Schlüssellänge von mehr als 64 Bit verwendet und darüber hinaus vom Anwen-

der „leicht“ zu modifizieren ist. PGP 2.6.x, die Freeware-Version von PGP 5.x/6.x und GnuPG bieten zwar denselben Sicherheitsstandard, sind als „public domain“⊘ aber von allen Regelungen des Abkommens ausgenommen. Einschränkungen des Bundesausfuhramtes über die Regelungen des Abkommens hinaus sind uns nicht bekannt.

6.3. Exportkontrolle in den USA

Die US-Regierung hat eine lange Geschichte von Exportregulierungen für kryptographische Technologie, die schon seit Jahrzehnten wie Kriegswaffen behandelt werden. Die Gesetze, die dabei bis vor kurzem zur Anwendung kamen, erweckten den Eindruck, als stammten sie aus einer Zeit, als ein Verschlüsselungsapparat noch mindestens die Größe eines Reisekoffers hatte, ihre Anwendung auf Software und insbesondere auf freie, nichtkommerzielle Software ist daher seit langer Zeit umstritten gewesen.

Wie bereits im vorangegangenen Abschnitt erwähnt, sind auch die USA am Wassenaar-Abkommen beteiligt; die Auswirkungen auf die nationale Gesetzgebung und die Entscheidungen des State Department bleiben abzuwarten. Die wirtschaftlichen Gründe für die US-Regierung, das Wassenaar-Abkommen zu initiieren,+ sind offensichtlich: Die einheimische Softwareindustrie hält sich – vermutlich zu recht – mit den bisherigen Exportbeschränkungen im internationalen Kontext für nicht wettbewerbsfähig, die US-Regierung sieht sich also wachsendem Druck ausgesetzt, die traditionellen Exportregulierungen neu zu überdenken. Die Ausnahmeregulierung für freie Software im Wassenaar-Abkommen ist Vertretern der USA aber anscheinend ein Dorn im Auge, auf politischer Ebene laufen Bestrebungen, diese Ausnahmeregelung aus dem Abkommen hinauszubefördern.

Wenn Sie außerhalb der USA und Kanadas leben, rate ich Ihnen, gegen die noch gültigen Verordnungen des State Department nicht dadurch zu verstoßen, daß Sie sich PGP aus den USA besorgen. Tausende von US-Bürgern haben sich PGP nach seiner ersten Veröffentlichung besorgt, und irgendwie ist es dann aus den USA herausgekommen und

⊘ Dieser Begriff wird im Abkommen explizit definiert als Software, die „ohne Einschränkungen der weiteren Verteilung zur Verfügung gestellt“ wurde. PGP 5.x/6.x ist keine public domain im üblichen Sinne.

\+ So sieht es zumindest für uns aus.

hat sich dann wie Unkraut von selbst weiterverbreitet. Wenn PGP bereits den Weg in Ihr Land gefunden hat, dann werden Sie wahrscheinlich keine US-Exportgesetze verletzen, wenn Sie sich PGP aus einer Quelle außerhalb der USA besorgen.

Die Versionen 2.0 bis 2.3a von PGP entstanden außerhalb der USA und wurden dort veröffentlicht, auf öffentlich zugänglichen Computern in Europa. Jede dieser Veröffentlichungen hat ihren Weg in die USA gefunden. Es gibt einige Beschränkungen in den USA, die die Einfuhr von Kriegswaffen reglementieren, aber diese sind unseres Wissens nie auf Software angewendet worden. Juristische Maßnahmen gegen einen solchen Import dürften eine spektakuläre Auseinandersetzung ergeben.

Einige Regierungen verhängen hohe Strafen allein für verschlüsselte Kommunikation. In Ländern, in denen Kriegsrecht gilt, kann man dafür erschossen werden (Verschlüsselung=Spion). Aber wenn man in so einem Land lebt, braucht man PGP vielleicht um so mehr. Im neuen Datenformat OpenPGP (vgl. Anhang A auf Seite 264) ist daraufhin sogar eine Möglichkeit eingebaut worden, eine Nachricht so zu verschlüsseln, daß sie keinen Hinweis darauf enthält, an wen sie verschlüsselt wurde – abgesehen natürlich davon, daß der berechtigte Empfänger sie entschlüsseln kann.

Falls Sie jemand darum bittet, unverschlüsselt zu senden, beachten Sie bitte diese Aufforderung *unbedingt*. In Kriegsgebieten kann allein das Empfangen einer verschlüsselten Nachricht für den Empfänger bedeuten, als Spion sofort und ohne gerichtliches Verfahren standrechtlich erschossen zu werden! Übrigens werden auch die Briefumschläge des Roten Kreuzes in Krisenregionen immer unverschlossen transportiert.

6.4. Das deutsche Signaturgesetz

Die deutsche Bundesregierung hat 1997 ein Gesetz verabschiedet, das einen rechtlichen Rahmen für elektronische Unterschriften im Rechtsverkehr schaffen soll. Nach Meinung einiger Fachleute ist das Gesetz stellenweise „mit der heißen Nadel gestrickt“,◁ was sich zumindest leicht erklären läßt, denn es handelt sich um das weltweit erste Gesetz dieser Art, und diese Vorreiterrolle ist natürlich sowohl den Politikern, als auch – und das dürfte ausschlaggebend sein – den beteiligten und be-

◁ MdB TAUSS zu Minister RÜTTGERS: „Herzlichen Glückwunsch, Sie haben gerade elektronisches Geld geschaffen.“ RÜTTGERS erstaunt: „Was habe ich?“

troffenen Industrien sehr wichtig. Wie dem auch sei, das Gesetz existiert und wird definitiv Auswirkungen auf das Leben in Deutschland haben. Außerdem hat es genug mit dem Thema dieses Buches zu tun, um ein paar Absätze darüber zu verlieren, auch wenn es mit PGP selbst nichts zu tun hat.

Das Signaturgesetz (SigG) definiert, welche Eigenschaften eine elektronische Unterschrift nach SigG haben soll, es legt aber nichts fest, was die Bedeutung dieser Unterschrift angeht. Derzeit werden Referentenentwürfe geschrieben, wie einige andere Gesetze abzuändern sind, um dort neben einer handschriftlichen auch eine digitale Unterschrift zuzulassen. Anders gesagt: Momentan hat eine Unterschrift nach SigG genau denselben Stellenwert wie jede andere digitale Signatur, unter dem Vorbehalt, daß dem einen oder anderen Richter eine Unterschrift nach SigG vermutlich als stichhaltigerer Anscheinsbeweis erscheint.

Was sind nun die Unterschiede zwischen einer Unterschrift nach SigG und den von PGP verwendeten? Zum Einen betrifft das das Datenformat – im Gesetzestext wird zwar kein Format festgelegt, es wird aber (in der zugehörigen Signaturverordnung) darauf verwiesen, daß das Bundesamt für Sicherheit in der Informationstechnik (BSI) eine entsprechende Spezifikation für die Zertifikate und Unterschriften erarbeiten soll. Diese liegt inzwischen vor und ist an X.509v3 angelehnt, das nur am Rande. Ein wesentlicher Unterschied liegt aber darin, daß ein Zertifikat nach SigG voraussetzt, daß die ausstellende Stelle (die ihrerseits von der Regulierungsbehörde zertifiziert sein muß) sicherstellt, daß vom privaten Schlüssel nur ein einziges Exemplar existiert – in der Praxis bedeutet das (und ist in der Signaturverordnung dann auch so festgelegt worden), daß der private Schlüssel auf einer vom BSI zertifizierten Chipkarte gespeichert sein muß und zwar so, daß er entweder von der Karte selbst erzeugt wurde oder vom ausstellenden Institut erstellt und auf die Karte aufgebracht wurde. Letzteres ist natürlich die kosteneffizientere Methode, da die Chipkarte die Fähigkeit, einen Schlüssel zu erzeugen, nur einmal braucht und es teuer ist, sie einzubauen. Gleichzeitig ist es aber auch die Methode, bei der die „Privatheit“ des privaten Schlüssels für den Kunden nicht kontrollierbar ist. Auch die Chipkartenleser, die dafür gedacht sind, daß dort Leute (nicht zu Hause am eigenen PC) digitale Unterschriften leisten, müssen vom BSI oder einer der wenigen anderen anerkannten Stellen zertifiziert sein. Abgesehen davon definiert das SigG – wie sein Name bereits sagt – nur Unterschriften, der Komplex der Verschlüsselung ist dort vollkommen ausgeklammert.

7. Ein paar Worte zum Umgang mit Schlüsseln

7.1. Öffentliche Schlüssel vor Manipulation schützen

PGP bietet ein Verschlüsselungssystem, das mit öffentlichen Schlüsseln arbeitet. Diese brauchen nicht vor Entdeckung geschützt zu werden. Im Gegenteil, je weiter sie verbreitet sind, desto besser. Andererseits ist es sehr wichtig, bei öffentlichen Schlüsseln sicherzugehen, von wem sie stammen. Hier liegt wahrscheinlich die größte Schwachstelle eines jeden Systems mit öffentlichen Schlüsseln. Lassen Sie uns ein kleines Desaster basteln und anschließend klären, wie sich derartige Situationen mit PGP vermeiden lassen.

7.1.1. Eine dumme Situation

Nehmen wir an, Sie wollen Alice eine eher private Nachricht schicken. Dazu holen Sie sich aus einer öffentlichen Quelle Alices öffentlichen Schlüssel, verschlüsseln die Nachricht damit und senden sie per E-Mail an Alice.

So weit, so gut, aber leider hat, was keiner wußte, Charlie eine Möglichkeit gefunden, um sich selbst auf dem Weg einzuhängen. (Das ist nicht wirklich schwierig.) Charlie hat also eine Möglichkeit, den gesamten E-Mail-Verkehr von und zu Alice abzufangen, zu verändern oder einfach mitzulesen. Er ersetzt den von Alice veröffentlichten Schlüssel durch einen öffentlichen Schlüssel, den er selbst erzeugt hat. Da dieser Schlüssel natürlich die ID von Alice trägt, verwenden Sie ihn, Charlie fängt die Nachricht ab, liest sie und schickt sie dann entweder – mit Alices wirklichem öffentlichen Schlüssel codiert – weiter, verändert sie eventuell zwischendurch, oder schickt sie überhaupt nicht weiter. Da er den privaten Schlüssel für den Schlüssel hat, von dem Sie glauben, er sei Alices, kann er dies alles tun. Alice erhielt natürlich ebenfalls einen falschen Schlüssel, als sie Ihren erhalten wollte, so daß auch das Unterschreiben von Nachrichten ohne Probleme zu fälschen ist und Alices Nachrichten an Sie ebenso abgefangen werden können.

7.1.2. PGPs Ausweg

Es gibt nur einen Ausweg aus diesem Dilemma: Öffentliche Schlüssel müssen vor derartigen Angriffen geschützt werden. Wenn Alice Ihnen ihren öffentlichen Schlüssel direkt in die Hand gedrückt hat, stellt das kein Problem dar, aber wenn Sie in der Schweiz wohnen und Alice in Norwegen, kann das etwas problematisch sein.

Vielleicht kennen Sie aber jemanden, dem Sie vertrauen und der zufällig gerade nach Norwegen in Urlaub fährt. Diesem netten Menschen könnten Sie dann ja Ihren öffentlichen Schlüssel mitgeben, und er würde Ihnen Alices bringen. Falls Alice den Menschen nicht zufällig ebenfalls gut kennt, kann sie zwar immer noch nicht sicher sein, daß der Schlüssel, den sie hat, von Ihnen ist, aber Sie können relativ sicher sein.

Nun ist es aber eher unpraktisch, zu allen E-Mail-Bekanntschaften Freunde hinzuschicken oder selber hinzufahren, um Schlüssel auszutauschen. Daher bietet PGP die Möglichkeit, Schlüssel/Benutzer-Zuordnungen zu „unterschreiben“.

Nehmen wir einmal an, Sie kennen außer Alice auch noch deren Halbbruder David. David hat Sie letzten Sommer besucht und bei der Gelegenheit seinen öffentlichen Schlüssel dagelassen. Unterschreibt David nun Alices öffentlichen Schlüssel, können Sie – wenn Sie ihm nicht zutrauen, daß er einen falschen oder zweifelhaften Schlüssel unterschreiben würde – davon ausgehen, daß der Schlüssel authentisch ist.

David würde also Alices öffentlichen Schlüssel unterschreiben, und Alice könnte diesen unterschriebenen Schlüssel versenden. Zwar kann Charlie den Schlüssel abfangen und durch seinen eigenen ersetzen, aber er kann nicht Davids Unterschrift unter diesen Schlüssel setzen. Selbst wenn er einen von David signierten Schlüssel hat, den David ihm unterschrieben hat, nützt ihm das nichts, da eine Unterschrift immer eine Zuordnung von ID und Schlüssel bestätigt.

Es wäre sogar möglich, daß eine weithin bekannte und allgemein als vertrauenswürdig eingestufte Person sich darauf spezialisiert, derartige Unterschriften zu leisten. Diese Person könnte also als „Beglaubigungsstelle“ arbeiten. Jeder, der sich an diesem Konzept beteiligen möchte, braucht nur den öffentlichen Schlüssel dieser Instanz (der natürlich auf extrem vertrauenswürdigen Kanälen zu ihm kommen muß) zu kennen und kann damit alle Unterschriften überprüfen.

Dieses Konzept ist vor allem für große, zentral gesteuerte Einrichtungen, wie Betriebe oder staatliche Verwaltungsapparate interessant. Hierzu ist vielleicht auch Abschnitt 6.4 über das deutsche Gesetz zu digitalen Unterschriften interessant. PGP zielt mehr auf ein dezentrales Beglaubigungssystem, in dem jeder Benutzer die Schlüssel derjenigen Leute, die er persönlich kennt oder die er auf Kongressen oder sonstwo trifft (und über deren Identität er sicher ist), unterschreibt. Das spiegelt eher den natürlichen Umgang mit Mitmenschen wider und bietet jedem die Möglichkeit, selbst zu entscheiden, wem er vertraut, wenn es um glaubwürdige Unterschriften geht.

Dieses Konzept ist wichtig!
Auch wenn es unwichtig erscheint: Diese Geschichte ist im Endeffekt die Achillesferse des ganzen Systems. Für vertrauliche Nachrichten (und wenn Sie längere Zeit E-Mail benutzen, werden Sie mit Sicherheit vertrauliche Nachrichten schreiben) sollten Sie niemals einen Schlüssel verwenden, von dem Sie nicht sicher sein können, von wem er stammt.

Von der Echtheit eines Schlüssels können Sie überzeugt sein, wenn Sie ihn direkt von seinem Besitzer bekommen haben, oder wenn er von einer Person unterschrieben ist, der Sie vertrauen, und von der Sie einen Schlüssel haben, der mit Sicherheit echt ist.

Daraus folgt auch:

- Unterschreiben Sie niemals einen Schlüssel, von dem Sie nicht absolut und hundertprozentig sicher sein können, daß er von der Person stammt, deren ID Sie unterzeichnen. Mit der Unterschrift bürgen Sie mit Ihrem guten Namen für die Echtheit des Schlüssels.
- Unterschreiben Sie keine Schlüssel, weil Sie jemanden gut kennen, der diese Schlüssel unterschrieben hat, sondern nur dann, wenn Sie wirklich selber unabhängig sicher sind und sein können, daß der Schlüssel echt ist. Auch die Schlüssel, die Sie mit PGP zusammen erhalten haben, sind nicht unbedingt echt, genauso, wie die digitalen Zertifikate in ihrem Internet-Browser theoretisch gefälscht sein könnten – nur würde es bei denen sehr viel schneller auffallen.
- Unterschreiben Sie vorzugsweise nur Schlüssel, die Sie direkt vom Besitzer erhalten haben. Eine Unterschrift unter einen Schlüssel zu setzen, setzt eine viel größere Sicherheit in punkto Eigentümerschaft voraus als das Verwenden eines Schlüssels.

Denken Sie auch daran, daß eine Unterschrift unter einem Schlüssel nichts darüber aussagt, wie vertrauenswürdig die Person ist, deren Schlüssel Sie unterschrieben haben, sondern daß die Unterschrift nur dafür bürgt, daß dieser Schlüssel wirklich zu der Person gehört. Einen Schlüssel als echt anzusehen ist eine Sache, dem Besitzer des Schlüssels zu vertrauen, eine andere. Mit PGP 6.0 haben Sie zusätzlich die Möglichkeit, in eine Unterschrift hineinzuschreiben, die betreffende Person sei vertrauenswürdig.

Vertrauen ist nicht unbedingt übertragbar: Nehmen wir an, ich vertraue einem Freund und bin mir sicher, daß er nicht lügt. Wenn er nun sicher ist, daß der US-Präsident nicht lügt, muß ich nicht notwendigerweise davon ausgehen, daß das stimmt. Wenn ich Davids Unterschrift unter Alices Schlüssel vertraue und David Alices Unterschrift unter Charlies Schlüssel vertraut, brauche ich noch lange nicht Charlies Schlüssel als echt anzusehen.

Daher ist es sinnvoll, unter seinem eigenen Schlüssel einigermaßen viele Unterschriften zu sammeln und den so signierten Schlüssel möglichst öffentlich zu verbreiten. Gute Methoden zum Verbreiten eines Schlüssels sind beispielsweise entsprechende Foren in Computernetzen (z. B. `/Z-NETZ/ALT/PGP/SCHLUESSEL`) oder die „public key server“ im Internet, z. B. `pgp-public-keys@informatik.uni-hamburg.de`. Wenn Sie einen Schlüssel unterschreiben, dann sollten Sie diesen mit Unterschrift an den Besitzer schicken.

PGP überwacht selbsttätig, welche Schlüssel in Ihrem Bund ausreichend mit Unterschriften bestätigt sind, die von Leuten geleistet wurden, die Sie als vertrauenswürdig eingestuft haben. Sie müssen PGP nur mitteilen, welche Leute Sie als vertrauenswürdig einstufen. Haben Sie selbst diese Schlüssel unterschrieben und somit für Ihr PGP voll bestätigt, können Sie mit Hilfe einer oder mehrerer Unterschriften unter einem Schlüssel, den Sie neu kriegen, diesen direkt als bestätigt einstufen lassen. Wie gesagt, vollautomatisch.

Weitere Informationen zum Unterschreiben fremder Schlüssel und den Folgerungen, die PGP aus diesen Unterschriften zieht, finden Sie im Abschnitt 7.3.

7.2. Die Schlüssel und andere Leute

Stellen Sie sicher, daß niemand außer Ihnen Zugriff auf Ihren Bund mit öffentlichen Schlüsseln hat. Die Überprüfung von Unterschriften hängt davon ab, welche öffentlichen Schlüssel bereits als vertrauenswürdig eingestuft sind. Daher sollten Sie nicht nur auf Ihren privaten Schlüssel aufpassen, sondern auch beim Umgang mit Ihrer privaten Sammlung öffentlicher Schlüssel Vorsicht walten lassen. Das heißt nicht unbedingt, die Datei davor zu schützen, daß sie gelesen werden kann, aber auch das kann Sinn machen. Schließlich geht es niemanden etwas an, wem Sie vertrauen.

Da der eigene öffentliche Schlüssel als Aufhänger aller Vertrauensketten derjenige ist, der am besten geschützt sein muß, bietet PGP die Möglichkeit, diesen mit einer Sicherheitskopie zu vergleichen, die auf einem schreibgeschützten Medium (Diskette mit Schreibschutz, Lochstreifen, CD) gespeichert sein sollte. Für nähere Informationen lesen Sie bitte Abschnitt 14.

PGP geht normalerweise davon aus, daß Sie Ihre Schlüssel sorgsam verwalten und vor herumstreunenden Kindern und Bösewichten schützen. Wenn jemand an Ihre Schlüsselbunde herankommt und sie manipulieren kann, wird er aller Wahrscheinlichkeit nach dasselbe mit dem PGP-Programm tun können und es so verändern, daß alle Schutzmechanismen wirkungslos sind.

Eine zugegebenermaßen eher umständliche Methode, die Sammlung öffentlicher Schlüssel vor Manipulation zu schützen, ist die, die Schlüsselbunde mit dem eigenen (privaten) Schlüssel zu unterzeichnen. Beispielsweise könnten Sie mit der Option `-sb` eine abgelöste Unterschrift für Ihren öffentlichen Schlüsselbund erzeugen (siehe Abschnitt 15.2). Diese bleibt natürlich nur so lange gültig, bis Sie einen neuen Schlüssel aufnehmen oder eine andere Änderung vornehmen.

Leider müßten Sie dann immer noch eine glaubwürdige Kopie Ihres eigenen Schlüssels zur Hand haben, um die Unterschrift prüfen zu können. Dem Schlüssel in der signierten Datei können Sie nicht vertrauen, da ein Mensch, der diesen Schlüssel fälscht, ebensogut eine neue Unterschrift erzeugen kann. Das ist viel einfacher, als einen Schlüssel an sich zu bringen.

7.3. Wie untersucht PGP, welche Schlüssel gültig sind?

PGP überprüft, welche öffentlichen Schlüssel Ihrer Sammlung als gültig eingestuft sind. Dazu müssen Sie PGP mitteilen, welchen Leuten Sie genug Vertrauen entgegenbringen, um ihre Unterschriften als gültige Bestätigungen für neue Schlüssel einzustufen. Hiervon ausgehend kann PGP neu hinzukommende Schlüssel direkt auf ihre Gültigkeit hin prüfen. Teilschlüssel eines als echt angesehenen Schlüssels, die mit dem Hauptschlüssel unterschrieben sind, gelten als echt.

PGP verwendet zwei verschiedene Kriterien, um Schlüssel einzustufen, bitte bringen Sie sie nicht durcheinander:

1. Gehört der Schlüssel wirklich der angegebenen Person?

2. Ist diese Person ausreichend glaubwürdig und vertrauenswürdig, andere Schlüssel zu bestätigen?

Die Antwort auf Frage 1 berechnet PGP, wobei Ihre Antworten auf Frage 2 für andere Benutzer zu Rate gezogen werden. Die Antwort auf die zweite Frage kann PGP nicht selbständig finden, sondern Sie müssen sie geben. Geben Sie bei Frage 2 für einige ausgewählte Benutzer als Antwort „Ja“ oder „Na ja, einigermaßen“, dann kann PGP Frage 1 für andere Schlüssel bejahen, die von diesen ausgewählten Benutzern unterschrieben wurden. Schlüssel, die von vertrauenswürdigen Personen unterschrieben wurden, werden von PGP als echt eingestuft. Um als vertrauenswürdig gelten zu können, muß ein Schlüssel von PGP als echt angesehen werden, also entweder von Ihnen selbst oder anderen vertrauenswürdigen Personen signiert sein – schließlich macht es keinen Sinn, einem Schlüssel zu vertrauen, von dem Sie nicht wissen, ob er der angegebenen Person wirklich gehört.

Sie müssen aber nicht allen Leuten gleich weit vertrauen. PGP bietet die Möglichkeit, verschiedene Stufen des Vertrauens zu vergeben. Ihre Einstufung einer Person in Bezug auf Glaubwürdigkeit sollte nicht nur Ihre Einschätzung der allgemeinen Zuverlässigkeit dieser Person widerspiegeln, sondern auch Ihre Einschätzung darüber, wie gut die Person das Prinzip der Unterschriften verstanden hat und wie ernst sie die damit verbundene Verantwortung nimmt. Sie können einer Person (genauer: Einem als echt eingestuften Schlüssel einer Person) die Vertrauensstufen „ich weiß nicht“, „nein“, „meistens ja“ oder „ja“ geben. Diese Information wird mit dem Schlüssel zusammen in Ihrer Sammlung gespei-

chert, aber wenn sie einen Schlüssel aus Ihrem Schlüsselbund herauskopieren, wird diese Angabe nicht mitkopiert. Ihre private Meinung geht schließlich niemanden etwas an, wenn Sie sie nicht ausdrücklich mitteilen wollen.

Wenn PGP einen Schlüssel auf seine Gültigkeit hin untersucht, zieht es die Vertrauenslevel aller Schlüssel zu Rate, mit denen dieser Schlüssel unterschrieben wurde. Sie können einstellen, wie viele Unterschriften von welcher Glaubwürdigkeit gebraucht werden, um einen Schlüssel zu bestätigen. In der Standardeinstellung hält PGP einen Schlüssel für echt, wenn er von einer mit „ja“ eingestellten Unterschrift geziert wird oder wenn er zwei Unterschriften mit der Einstellung „meistens ja“ trägt.

Der eigene öffentliche Schlüssel gilt definitionsgemäß als vertrauenswürdig und echt, seine Unterschrift reicht immer aus, um einen Schlüssel als echt zu bestätigen. GnuPG gestattet es, diesen Status auch anderen Schlüsseln zu geben, für den Fall, daß Sie beispielsweise einer Zertifizierungsinstanz das nötige Vertrauen entgegenbringen. PGP 5.0 bietet die Möglichkeit, einem Schlüssel die Eigenschaft „exportable metaintroducer“ zu verleihen, was im Wesentlichen bedeutet: „Vertraue diesem Schlüssel und laß andere wissen, daß ich ihm vertraue.“ Die Vertrauenseinstellung wird also mit exportiert. PGP 6.0 nennt diese Einstellung „trusted introducer“, hier ist die Einstellung „meta introducer“ neu hinzugekommen, die besagt, daß jeder Schlüssel, der von diesem Schlüssel als vertrauenswürdig gekennzeichnet ist, als vertrauenswürdig angesehen werden soll.

Mit der Zeit werden Sie einige Schlüssel sammeln, die Sie als mehr oder weniger vertrauenswürdig einstufen. Andere Leute, die teilweise dieselben Schlüssel haben, werden andere Einschätzungen vergeben. Mit der Zeit werden alle Benutzer ihre Schlüssel mit immer mehr Unterschriften verteilen. Dadurch ist das so gesponnene Vertrauensnetz sehr viel weniger gefährdet, durch eine Stelle zu zerreißen, die schwächer ist, als vorher angenommen.

Dieses dezentrale Konzept ist nicht das einzig mögliche. Es gäbe auch die Möglichkeit, Schlüssel von einer zentralen Instanz unterschreiben zu lassen, und bei einigen verwandten Verfahren wird das auch gemacht, beispielsweise bei Privacy Enhanced Mail (PEM). Dieses Schema basiert auf einer vorgegeben Hierarchie, in der andere Leute festlegen, wem Sie zu vertrauen haben. PGP geht den anderen Weg, bei dem Sie diese Entscheidung selber fällen müssen.

PGP ist für Leute, die ihren Fallschirm selbst zusammenlegen.

7.4. Private Schlüssel vor Diebstahl schützen

Schützen Sie sowohl Ihren privaten Schlüssel als auch Ihr Mantra sorgfältig. Sollte Ihr privater Schlüssel jemals das Licht der großen, weiten Welt erblicken oder auch nur mit einem Anderen fremdgehen, dann sollten Sie möglichst schnell möglichst vielen Leuten davon erzählen, nach Möglichkeit allen, die den öffentlichen Schlüssel haben. Viel Glück dabei. Möglichst bevor irgend jemand Unsinn damit anstellt, wie z. B. Schlüssel damit zu unterschreiben oder in Ihrem Namen auf Nachrichten zu antworten.

Das erste, was Sie tun sollten, um Ihren Schlüssel zu schützen, ist, den Schlüssel selbst immer unter Kontrolle zu haben. Das heißt, Sie sollten ihn nicht auf einem Computer lassen, wo er dem Zugriff anderer Personen ausgesetzt ist. Wenn Sie keine Möglichkeit dazu haben, weil sie beispielsweise Ihren gesamten E-Mail-Verkehr von einer Maschine im Büro aus abwickeln, sollten Sie Ihren Schlüssel immer auf einer schreibgeschützten Diskette mit sich tragen und von da aus arbeiten. Vergessen Sie dabei Ihre Sicherheitskopie nicht. Einen privaten Schüssel auf einem solchen System, insbesondere auf einem vernetzten Rechner, aufzubewahren, ist nicht gerade sinnvoll. Und Sie sollten PGP erst recht nicht über ein unverschlüsseltes Netzwerk, z. B. eine Modemverbindung benutzen. Es wäre zu einfach, Ihr Mantra mitzuschneiden.

Außerdem sollten Sie natürlich beachten, was für Paßwörter allgemein gilt: Sie sollten Ihr Mantra auf gar keinen Fall irgendwo aufschreiben und am Monitor, unter der Tastatur oder sonst irgendwo an Ihrem Arbeitsplatz, in der Aktentasche oder an welchem Ort auch immer aufbewahren, an den jeder potentielle Eindringling gelangen kann. Das Sicherste ist immer noch, es einfach im Kopf zu behalten.

Bewahren Sie außerdem immer eine Sicherheitskopie Ihres privaten Schlüssels auf. Wenn Ihre Festplatte einmal den Geist aufgibt und Sie ohne Ihren privaten Schlüssel dastehen, können Sie die gesamte an Sie geschickte Post nicht mehr lesen.

Hier kommen wir bei einem Problem der dezentralen Methode zur Schlüsselverifikation an: Da es keine Zentrale gibt, die Schlüssel bestätigt, gibt es auch keine, die vor unsicher gewordenen, da möglicherweise in fremde Hände gefallenen Schlüsseln, warnt. Sie können nur die Meldung, daß Ihr Schlüssel möglicherweise bekannt wurde, möglichst weit verbreiten und darauf hoffen, daß diese Nachricht auch bei allen Leuten ankommt, die davon erfahren sollten.

Sollte der Fall der Fälle eintreten, sollte also sowohl Ihr privater Schlüssel als auch Ihr Mantra in fremde Hände fallen, dann sollten Sie eine sogenannte Rückrufurkunde („key compromise certificate“) ausstellen. Dies ist eine Urkunde, die besagt, daß Ihr Schlüssel auf keinen Fall mehr verwendet werden darf. Näheres hierzu finden Sie in Abschnitt 13.2.8 ab Seite 87.

7.5. Ich habe meinen privaten Schlüssel verloren, was jetzt?

Normalerweise lassen sich Schlüssel zurückziehen. Da hierbei aber eine Unterschrift nötig ist, kann der Befehl nicht mehr aufgerufen werden, wenn der private Schlüssel verlorengeht. Also, was dann? Sie können Ihren Schlüssel nicht mehr auf dem oben genannten Weg zurückziehen, Sie können aber auch keine mit dem öffentlichen Schlüssel verschlüsselten Nachrichten lesen. Versionen ab 6.0 bieten einen Mechanismus, der den Schlüssel mit Hilfe von als glaubwürdig eingestuften Personen[⊤] unbrauchbar macht. Bei Verwendung der früheren Versionen bleibt Ihnen wohl keine andere Möglichkeit als die, alle User zu bitten, Ihren Schlüssel nicht mehr zu verwenden. Diese Nachricht sollten Sie von denselben Leuten unterschreiben lassen, die auch Ihren Schlüssel signiert haben.

⊤ d. h., Sie können in Ihrem Schlüsselzertifikat vermerken, wer diesen Schlüssel zurückziehen darf.

Teil II.

Kommandozeilenversionen
PGP 2.6.x, 5.x, GnuPG

8. Überblick

In diesem Teil des vorliegenden Buches finden Sie eine Bedienungsanleitung für die Kommandozeilenversionen von PGP. Für viele Einsatzzwecke (automatisierte Verschlüsselung von Logfiles/Backups z. B.) ist das die einzige Einsatzmöglichkeit (eine graphische Benutzerschnittstelle läßt sich nur schwer von einer Stapeldatei aus bedienen), außerdem ist die Bedienungsschnittstelle systemunabhängig. Sie können die hier beschriebenen Kommandos fast ohne Änderungen auf Unix, Linux, Windows, MS-DOS, Amiga, VAX/VMS, Atari und so weiter verwenden, lediglich für den Macintosh scheint es keine derartigen Versionen zu geben.

Um Einsteigern ein Geleit an die Hand zu geben, möchten wir hier eine mögliche Reihenfolge vorschlagen. Wir gehen dabei davon aus, daß Sie die in 1.1 auf Seite 2 genannten Kapitel gelesen haben.

Als erstes steht sicherlich die Installation PGPs an. Welchem Kapitel Sie die entsprechende Anleitung entnehmen können, ist abhängig davon, ob Sie das immer noch beliebte PGP 2.6.2i (Kapitel 10 auf Seite 72), die Freeware-Version von PGP 5.0i (Kapitel 11 auf Seite 74) oder die freie Software GnuPG (Kapitel 12 auf Seite 76) installieren möchten. Diese Entscheidung ist nicht unbedingt leicht zu treffen, wir können Ihnen nur Anhaltspunkte geben. Wenn Ihre Kommunikationspartner bereits PGP-Versionen 5.0 oder höher einsetzen, werden Sie vermutlich wenig Freude an PGP 2.6.2i haben. Um mit Menschen zu kommunizieren, die PGP 2.6.2i oder auch in den neueren Versionen RSA-Schlüssel einsetzen, ist GnuPG leider nicht geeignet, da es weder RSA noch IDEA unterstützt. Auf der anderen Seite darf die Freeware-Version von PGP 5.0i nicht kommerziell eingesetzt werden; für GnuPG gibt es keine derartigen Einschränkungen. Benötigen Sie Features der Version 2.6.2i wie das „Verschlüsseln nur für die Anzeige am Bildschirm“ (vgl. Abschnitt 15.9 auf Seite 130), das physikalische Löschen Ihrer Dateien (Abschnitt 5.4 auf Seite 37) oder auch das nachträgliche Trennen von Unterschrift und Nachricht (Abschnitt 15.2 auf Seite 125), brauchen Sie logischerweise diese Version.

Haben Sie PGP (erfolgreich) installiert, ist der nächste Schritt die Erzeugung eines eigenen Schlüsselpaars. Die Anleitung dazu finden Sie in Abschnitt 13.2.1 auf Seite 80. Mit diesem frisch erzeugten Schlüssel können Sie schon ein wenig experimentieren, um sich an die Bedienung PGPs oder des von Ihnen gewählten Frontends zu gewöhnen. Wenn Sie PGP 2.6.2i verwenden (nicht 2.6.3i oder später, dort wird der folgende Schritt automatisch durchgeführt), sollten Sie als erstes Ihren eigenen Schlüssel unterschreiben. Eine Anleitung dazu finden Sie in Abschnitt 13.2.7 auf Seite 86. Bei Ihrem eigenen Schlüssel können Sie die Frage nach der Authentizität wohl getrost bejahen. Mit den in Abschnitt 13.2.4 auf Seite 84 beschriebenen Befehlen sollten Sie jetzt Ihren neuen Schlüssel angezeigt bekommen.

Sie können nun ein paar Dateien an sich selbst verschlüsseln und wieder entschlüsseln, die nötigen Arbeitsschritte sind in den Abschnitten 13.3 auf Seite 88, 13.8 auf Seite 92 und 15.8 auf Seite 130 beschrieben. Probieren Sie auch das in Abschnitt 13.5 auf Seite 90 beschriebene Unterschreiben einer Nachricht aus; wenn Sie eine unterschriebene Nachricht testweise ändern wollen (um zu sehen, was PGP bei einer fehlgeschlagenen Unterschrift anzeigt), sollten Sie entweder eine Klartext-Unterschrift (Abschnitt 13.11 auf Seite 96) oder eine abgetrennte Unterschrift (siehe 15.2 auf Seite 125) verwenden – eine „normale“ unterschriebene Nachricht direkt zu verändern, führt bestenfalls dazu, daß PGP einen Fehler beim Dekomprimieren meldet.

Wenn von Ihren Bekannten noch niemand PGP einsetzt, sollten Sie ein weiteres Buch kaufen und es verschenken.

Um einem anderen PGP-Anwender Ihren öffentlichen Schlüssel zukommen zu lassen, müssen sie ihn mit den in Abschnitt 13.2.5 auf Seite 85 beschriebenen Funktionen in eine Datei exportieren, die Sie anschließend per Mail, Diskette oder wie auch immer übertragen können. Die Empfängerin verwendet dann einen der Aufrufe aus Abschnitt 13.2.2 auf Seite 83, um den öffentlichen Schlüssel einzulesen. Sinnvollerweise führen Sie diesen Schlüsseltausch in beide Richtungen aus. Anschließend können Sie Nachrichten aneinander verschlüsseln und sich unterschriebene Nachrichten zukommen lassen. Vorher sollten Sie natürlich die ausgetauschten Schlüssel kontrollieren. Da ihre Schlüssel bislang nur die eigenen Unterschriften tragen, braucht PGP dafür Ihre Hilfe. Der Vorgang ist in den Abschnitten 7.1.2 auf Seite 60 und 13.2.7 auf Seite 86 beschrieben.

9. Allgemeines

Die Versionen 2.6.x haben nur eine Kommandozeilenbedienung, es gibt aber eine Vielfalt von „Frontends“, die eine Bedienung per graphischer Oberfläche erlauben. Auf sie im Einzelnen einzugehen, würde den Rahmen dieses Buches bei weitem sprengen. Wir haben für Sie auf der beiliegenden CD einige Frontends im gleichnamigen Verzeichnis versammelt. Bitte lesen Sie die dortige Datei `README` sowie die jeweilige Programmdokumentation für genauere Informationen.

Die Versionen 5.x und folgende wurden explizit primär für Windows entwickelt und normalerweise per graphischer Oberfläche bedient; die Beschreibungen dieser Versionen finden Sie in Teil III ab Seite 144. Es gibt aber auch eine Kommandozeilenversion von 5.0, die in diesem Abschnitt beschrieben ist; für Unix ist das die einzige Version von PGP 5.x.

Wie Sie in Anhang A auf Seite 264 lesen können, hat es mit der Einführung der Version 5.0 neue Datenformate gegeben und es werden andere Algorithmen verwendet, die die Version 2.6.2 nicht verarbeiten kann. Ebenfalls in dem genannten Abschnitt finden Sie Hinweise darauf, warum viele Leute (unter anderem wir) „Bauchschmerzen“ beim Einsatz von PGP 5.x/6.x haben. Aus diesem Grund beschreiben wir in diesem Kapitel auch die Bedienung von GnuPG („Gnu Privacy Guard“), einer Implementation der neuen Algorithmen und Datenformate, die mit PGP 5.x/6.x kommunizieren kann – aber leider nicht mit PGP 2.6.x, da die dort verwendeten Algorithmen patentiert sind (vgl. Anhang E) und der Autor des Programmes sie nicht mit aufnehmen möchte.

Kurz vor Drucklegung dieses Buchs ist die Version 6.5.1 und die Version 6.5.1i PGPs erschienen. Die Quelltexte zu 6.5.1i lagen uns leider nur zur Vorabversion Beta-1 vor; sie ließen sich erst nach Änderungen überhaupt kompilieren und auch dann nur als Debug-, also Entwicklerversion. Die Zeit reichte nicht aus, um diese Version (und auch die vorkompilierte Version 6.5.1) auf ihre Stabilität zu testen. Die Bedienung der genannten Versionen unterscheidet sich nur geringfüfig von der der Versionen 2.6.x, die meisten Ändrungen betreffen neue Funktionen wie den

Zugriff auf Keyserver. Sofern bei einem Kommando nichts anderes angegeben ist, können Sie die für 2.6.x angegebene Befehlssyntax auch für den entsprechenden Aufruf bei 6.5.1(i) verwenden.

10. Die Installation von PGP 2.6.x

10.1. Allgemein

10.2. MS-DOS/Windows

Die MS-DOS-Version von PGP 2.6.2i ist in einer komprimierten Archiv-Datei namens `pgp262i.zip` enthalten. Diese Datei kann beispielsweise mit dem MS-DOS Sharewareprogramm `PKunzip` oder unter Windows mit `WinZIP` ausgepackt werden. Die Quelltexte finden Sie in der Datei `pgp262is.zip`.

Für die Installation wird `pgp262i.zip` einfach in ein geeignetes Verzeichnis auf der Festplatte kopiert (z. B. `c:\pgp`) und mit `PKunzip` ausgepackt. Um die Installation bequem nutzen zu können, sollten Sie auch die `autoexec.bat` ändern, und zwar sollte die Zeile

```
SET PGPPATH=c:\pgp
```

eingefügt werden (vgl. Abschnitt 13.12 auf Seite 97), unter Umständen ist auch noch

```
SET TZ=MET-1DST
```

nötig, näheres hierzu finden Sie in Abschnitt 14 auf Seite 119. Dies kann aber auch später erfolgen, wenn Sie sich ein wenig mit PGP beschäftigt und dieses Handbuch durchgelesen haben.

Das deutsche Sprachkit, bestehend aus den Dateien `config.txt`, `language.txt`, `de.hlp`, `readme.de` und `setup.de`, befindet sich in einer Datei namens `pgp-germ.zip`. Der Sprachkit sollte nach dem eigentlichen Programmpaket installiert werden, damit die Dateien `language.txt` und `config.txt` in ihrer deutschen Version auf der Festplatte vorhanden sind.

10.3. Linux

Für Linux-Systeme basierend auf RedHat oder SuSE finden Sie auf der CD die Datei `pgp-2.6.3i-1.i386.rpm`, die Sie mit `rpm`, `yast` oder vielen weiteren Installations-Utilities einspielen können. Das deutsche Sprachkit wird hierbei gleich mit installiert.

10.4. Unix/Andere

Die Installation unter Unix und VAX/VMS unterscheidet sich kaum von der MS-DOS-Installation, jedoch muß möglicherweise der Quellcode neu compiliert werden. Hierfür ist ein Unix-Makefile dem Quellcode beigefügt.

11. Die Installation von PGP 5.0

11.1. MS-DOS/Windows

Entpacken Sie das Archiv `5.0i/MS-DOS/pgp50ibi.zip` von der beiliegenden CD in das Verzeichnis, wo Sie PGP 5.0i installieren möchten (beispielsweise `c:\pgp`). Rufen Sie anschließend in diesem Verzeichnis

```
install f
```

auf und nehmen Sie das Verzeichnis in Ihrer `autoexec.bat` in die `PATH`-Variable auf. Außerdem sollten Sie die Variable `PGPPATH` setzen. Auf den meisten Systemen genügt es hierfür, an das Ende der `autoexec.bat` die folgenden Zeilen anzufügen:

```
set PGPPATH=c:\pgp
set PATH=%PATH%;%PGPPATH%
```

11.2. Linux

Sie können die Datei `5.0i/Unix/pgp-5.0i-1.i386.rpm` mit `rpm`, `yast` oder einer Reihe anderer Programme installieren. Der Aufruf für die Intallation per `rpm` (den Sie als User `root` ausführen müssen) lautet so:

```
rpm -Uvh /mnt/cdrom/5.0i/Unix/pgp-5.0i-1.i386.rpm
```

Alternativ können Sie wie im nächsten Abschnitt beschrieben vorgehen.

11.3. Unix

Wechseln Sie in in ein Verzeichnis, in dem Sie Quelltexte entpacken, beispielsweise `/usr/src`. Dort rufen Sie folgende Befehle auf:

```
gzip -dc /cdrom/5.0i/Unix/pgp50i-unix-src.tar.gz | tar xf -
cd pgp50i/src
./configure
make
```

und als User `root`

```
make install
```

12. Die Installation von GnuPG

12.1. Allgemein

Im Gegensatz zu PGP wertet GnuPG die Standard-Umgebungsvariable `LANG` aus. Um deutschsprachige Meldungen zu erhalten, sollten Sie also in ihrer Umgebung diese Variable auf `de` setzen.

12.2. Linux

Sie können das Paket `gnupg-1.0.0-1.i386.rpm` von der CD mit `rpm` oder einem beliebigen Frontend (`yast`, `glint` etc.) installieren. Alternativ können Sie auch wie im nächsten Abschnitt beschrieben vorgehen.

```
rpm -Uvh /mnt/cdrom/GnuPG/gnupg-1.0.0-1.i386.rpm
```

12.3. Unix

Sie finden die Quelltexte samt `Makefile` auf der beiliegenden CD als `GnuPG/gnupg-1.0.0.tar.gz`. Die Installation gestaltet sich sehr einfach. Wechseln Sie in ein Verzeichnis, in dem die Quellen ausgepackt werden sollen (z. B. `/usr/src`) und rufen Sie dort auf:

```
gzip -dc /mnt/cdrom/GnuPG/gnupg-1.0.0.tar.gz | tar xvf -
cd gnupg-1.0.0
./configure
make
su -c 'make install'
```

12.4. Windows 95/98/NT

Die Windows-Version von GnuPG auf der CD zu diesem Buch hat einen gravierenden Nachteil: Der Autor rät davon ab, den erzeugten Zufalls-

zahlen allzusehr zu vertrauen. Inwieweit das übetriebene Paranoia ist, sei einmal dahingestellt – vom Prinzip her ist das Vorgehen besser als das der kommerziellen PGP-Versionen.

Wenn Sie diese Version dennoch verwenden wollen, sollten Sie zunächst die Datei `GnuPG/gnupg-w32-1.0.0c.zip` in ein neues Verzeichnis entpacken. Anschließend sollten Sie die Datei `gpg.exe` an eine Stelle legen, wo Sie üblicherweise ausführbare Dateien haben. Auf einem frisch installierten Windows 98-System bietet sich beispielsweise das Verzeichnis `c:\windows\command\` an.

Legen Sie als nächsten Schritt das Verzeichnis `c:\gnupg` an und verschieben Sie die Datei `entropy.dll` dorthin.

Damit ist GnuPG fertig installiert. Ob und wie in der derzeitigen alpha-Version die mitgelieferten Sprachdateien eingebunden werden können, entzieht sich unserer Kenntnis.

13. PGP bedienen

Im Verlauf dieses Kapitels werden wir von den Versionen PGP 2.6.2i, PGP 5.0 und GnuPG 1.0.0 ausgehen.• Abweichungen zu anderen Versionen der Reihe 2.6.x finden Sie bei den entsprechenden Befehlen erwähnt. Näheres zu den verschiedenen PGP-Versionen finden Sie in Anhang A. Am Ende dieses Kapitels, ab Seite 99, finden Sie eine Kurzübersicht der besprochenen Befehle.

Texte, die in [eckigen Klammern] stehen, sind optional, können also weggelassen werden. Werden optionale Parameter angegeben, dürfen die eckigen Klammern selbst nicht angegeben werden. Wenn Zeilen zu lang für die Textbreite sind, werden sie in diesem Handbuch mit \ beendet und auf der nächsten Zeile fortgesetzt. Eine Kommandozeile der Art

```
beispiel: beispiel -a [-b] -c \
          -d -e -f [-g]
```

könnte also als

```
beispiel -a -c -d -e -f -g
```

eingetippt werden. (Unter Unix können Sie auch die Form mit \ am Ende der ersten Zeile, gefolgt von <Return> eintippen.)

13.1. Kurzanleitung am Bildschirm

Mit dem Kommando

```
2.6.x: pgp -h
  5.0: pgpe -h
  5.0: pgpk -h
  5.0: pgps -h
```

• Die Beispielausgaben sind teilweise mit anderen Versionen erzeugt worden. Wesentliche Unterschiede treten aber nicht auf.

5.0: `pgpv -h`
gpg: `gpg --help`

gibt PGP einen „Kurzüberblick“ über die möglichen Befehle.

Für 2.6.x gilt des Weiteren Folgendes: Abhängig davon, ob eine Datei namens `pgp.hlp` vorhanden ist oder nicht, wird der Befehl unterschiedlich ausgeführt. Ist `pgp.hlp` nicht vorhanden, erscheint eine etwa 15 Zeilen umfassende „Superkurzanleitung“ am Bildschirm. Wenn die Datei `pgp.hlp` vorhanden ist, wird ihr Inhalt am Bildschirm angezeigt. Im Text kann mit den Tasten `LEERTASTE`, `RETURN` und `B` geblättert werden. Mit `Q` wird das Programm beendet. Falls die Datei `de.hlp` vorhanden und in der Datei `config.txt` der Eintrag `language=de` enthalten ist, wird der Hilfstext auf deutsch angezeigt. Details zu `config.txt` finden Sie in Abschnitt 14. (Die hier angegebenen Tasten gelten natürlich nur, wenn in der `config.txt` kein spezielles Anzeigeprogramm eingestellt wurde.)

GnuPG verwendet kein internes Anzeigeprogramm und gibt eine kurze Hilfe einfach auf der Standardausgabe aus. Wenn Sie sie auf einem Bildschirm mit weniger als 83 Zeilen lesen möchten, sollten Sie eines der Kommandos

gpg: `gpg --help | less`
gpg: `gpg --help | more`

aufrufen.

13.2. Die Schlüsselverwaltung

Schon zu Zeiten Julius Cäsars war der Umgang mit Schlüsseln die delikateste Angelegenheit der Kryptographie. Einer der größten Vorteile von PGP ist die hochentwickelte Schlüsselverwaltung. GnuPG hat hier allerdings eine etwas andere Herangehensweise als PGP, für die meisten Befehle rufen Sie `gpg --edit-key User_ID` auf und geben dann einen passenden Befehl ein. In diesem Abschnitt finden Sie die entsprechenden Befehle so formatiert (aus Platzgründen beschränken wir uns auf „Menü“ statt „GnuPG-Menü“ o. ä.):

Menü: `fpr`

Der entsprechende Programmaufruf könnte wie folgt aussehen, wobei die kursiv gedruckten Teile die Benutzereingaben sind:

```
[ccr@nescio doc]$ gpg --edit-key Koch
gpg (GnuPG) 0.9.1; Copyright (C) 1999 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.


pub  1024D/57548DCD  created: 1998-07-07 expires: 2002-12-29 trust: -/q
(1)  Werner Koch (gnupg sig) <dd9jn@gnu.org>

Command> fpr
pub  1024D/57548DCD 1998-07-07 Werner Koch (gnupg sig) <dd9jn@gnu.org>
             Fingerprint: 6BD9 050F D8FC 941B 4341  2DCC 68B7 AB89 5754 8DCD

Command> quit
[ccr@nescio doc]$
```

13.2.1. Einen Schlüssel generieren

Um Ihr eigenes Schlüsselpaar zu erzeugen, geben Sie folgenden Befehl ein:

```
 2.6.x: pgp -kg
   5.0: pgpk +fastkeygen=0 -g
   gpg: gpg --gen-key
Menü: addkey
```

Nach der Eingabe dieses Befehls fragt PGP nach der Schlüsselgröße. PGP bietet einige mögliche Schlüsselgrößen als Auswahlmöglichkeiten an, 384 Bit „für den Hausgebrauch“, 512 Bit „für normale Anwendungen“ oder 1024 Bit „für militärische Sicherheit“, Sie können aber auch direkt z. B. 1536 eingeben. Die Werte für die Schlüsselgrößen sind allerdings nur als Anhaltspunkte anzusehen. Je größer der Schlüssel ist, desto sicherer ist die Verschlüsselung, aber auch umso langsamer. Die meisten Anwender verwenden 2048 oder 1024 Bit.

GnuPG fragt zunächst, welche Sorte Schlüssel Sie verwenden möchten. Da PGP 5.x/6.x von den angebotenen Schlüsseln nur die Kombination DSA/ElGamal (und RSA, allerdings nicht in allen Programmversionen) unterstützen, ist diese Kombination empfehlenswert, um mit Anwendern dieser PGP-Versionen Nachrichten austauschen zu können.

Des weiteren haben Sie die Möglichkeit, Schlüssel mit begrenzter Gültigkeitsdauer zu erzeugen; das ist nicht nur für zeitlich begrenzte Projekte interessant, sondern auch für den besonders sicherheitsbe-

wußten Einsatz zu empfehlen – wenn Sie die von Ihnen verwendeten Schlüssel regelmäßig wechseln, machen Sie einen Angriff schwerer und uninteressanter.

Der Schlüssel-Befehl `addkey` ist interessant, um an einen bestehenden (eigenen) Schlüssel einen neuen Teilschlüssel zu hängen, beispielsweise den Schlüssel, mit dem Sie in den nächsten sechs Monaten ausgehende Post unterschreiben wollen.

Anschließend möchte PGP eine User-ID wissen, d. h. einen Namen, der angibt, wem der Schlüssel eigentlich gehören soll. Vorzugsweise sollte der gesamte Name verwendet werden, da so weniger Verwechslungsmöglichkeiten entstehen. Es ist allgemeiner Usus, hinter dem Realnamen in <spitzen Klammern> eine E-Mail-Adresse anzugeben, also z. B.:

```
Patrick Lenz <MASTER@aol.com>
Johanna Wiesmüller <smb4423@rz.uni-sonstwo.de>
```

Für GnuPG wird diese Form bereits dadurch erzwungen, daß Sie Ihren Namen, Ihre E-Mail-Adresse und evtl. noch einen Kommentar (z. B. „nur für Pressemeldungen“) einzeln eingeben müssen. Diese Angaben werden im neuen Datenformat voneinander getrennt abgespeichert.

Nun fragt das Programm Sie nach einem Mantra, mit dem Ihr privater Schlüssel geschützt werden soll. Hierbei handelt es sich um die größere Ausgabe eines Paßwortes, die beliebig lang sein darf, also beispielsweise ein kompletter Satz. Ohne dieses Mantra ist Ihr privater Schlüssel praktisch wertlos. Dieses Mantra dürfen Sie auf keinen Fall vergessen, sonst haben Sie *keine* Möglichkeit, wieder an Ihren privaten Schlüssel zu kommen. Andererseits sollte das Mantra auch nicht zu leicht zu raten sein, denn es ist der einzige Schutz Ihres privaten Schlüssels vor Mitmenschen, die Daten von Ihrem Rechner herunterkopieren können.

Außerdem gilt das Übliche: Das Mantra nicht aufschreiben, kein kurzes oder anderweitig leicht zu ratendes Mantra verwenden („Alea iacta est“ wäre viel zu simpel, auch „Ich liebe Moni!“ sollte nicht verwendet werden), und das Mantra nicht über ein Netzwerk eintippen. Normalerweise erscheint das Mantra nie am Bildschirm, es sei denn, die Konfiguration wurde entsprechend geändert.

PGP unterscheidet beim Mantra zwischen Groß- und Kleinschreibung. Neben Buchstaben kann das Mantra auch Ziffern, Satzzeichen usw. enthalten. Sollten Sie es tatsächlich vorziehen, kein Mantra zu verwenden, drücken Sie einfach `RETURN`. Das halten wir aber für eine sehr schlechte Idee.

Um das Schlüsselpaar zu erzeugen, braucht PGP große, wirklich zufällige Zufallszahlen.Diese werden bei PGP 2.6.x aus den Zeitabständen zwischen einigen Tastendrücken abgeleitet, um die Sie gebeten werden. Tippen Sie einfach ein wenig Zufallstext in ständig wechselnder Geschwindigkeit. PGP sagt Ihnen, wieviel Sie tippen müssen. GnuPG ist darauf ausgelegt, vom System bereitgestellte Zufallszahlen zu verwenden (Linux: `/dev/random`).PGP 5.0i nutzt diese Quelle auf Linux ebenfalls. Diese Zufallszahlen stammen aus den Zeitangaben, zu welcher Millisekunde Sie welche Taste gebraucht haben, wieviele Millisekunden der Zugriff auf eine Datei Ihrer Festplatte diesmal gebraucht hat, wie sehr die internen Rechnerzeiten voneinander abweichen und derartigen (aller Wahrscheinlichkeit nach wirklich zufälligen) Angaben. Für Systeme, auf denen kein derartiger Mechanismus vorgesehen ist, wurde der „entropy gathering daemon“ entwickelt, der versucht, derartige Informationen im Hintergrund zu sammeln. Besser ist eine Integration in das System. Die nächste Generation der Intel-Chips, Pentium III, soll einen Hardware-Generator für Zufallszahlen enthalten. Sofern der Zugriff hierauf nicht durch das System eingeschränkt wird, wird das vermutlich eine gute *zusätzliche* Quelle zur Generierung echter Zufallszahlen sein.

Anschließend zeigt PGP Ihnen mit einigen `'.'` und `'+'` (GnuPG auch noch `'!'`, `'^'`, `'<'`, `'>'`) seinen Fortschritt an. Auf langsameren Rechnern kann das Ganze durchaus mehrere Minuten dauern, aber es wird im Allgemeinen ja nur selten (bei jedem Schlüsselwechsel, beispielsweise alle sechs Monate einmal) gemacht. Falls Sie genauer wissen wollen, wann welches Zeichen angezeigt wird: Das Programm braucht zur Schlüsselerzeugung große, geheime Primzahlen. Diese werden berechnet, indem bei einer Zufallszahl begonnen wird und dann so lange die nächste ungerade Zahl getestet wird,[▽] bis eine Primzahl gefunden wurde. Die Bedeutung der einzelnen Zeichen finden Sie in Tabelle 13.1. Die verwendeten Primzahltests (Rabin-Miller und Fermat) sind Zufallstests, die durchaus auch fehlerhaft behaupten können, eine Zahl sei prim, aber nicht umgekehrt – deswegen muß eine Zahl mehrere Durchläufe bestehen, um als Primzahl angesehen zu werden. Die Wahrscheinlichkeit dafür, daß PGP eine Zahl fehlerhaft als prim ansieht, ist verschwindend klein (kleiner als 2^{-16}).

▽ Die Suche nach einer Primzahl für ElGamal ist ein wenig komplizierter, aber das ist ein technisches Detail, das die Sicherheit verbessert.

Zeichen	Bedeutung
.	10 Tests sind fehlgeschlagen.
+	Eine Zahl hat einen Testdurchlauf bestanden.
!	Es werden neue Zufallsparameter gewählt.
^	Ein neuer Wert für den ElGamal-Parameter g wird getestet.
<	Die Größe eines der RSA-Faktoren wurde verkleinert.
>	Die Größe eines der RSA-Faktoren wurde vergrößert.

Tabelle 13.1: Bedeutung der Zeichen bei der Schlüsselerzeugung

Die frisch erzeugten Schlüssel werden vom Programm in den Bund mit öffentlichen bzw. den mit geheimen Schlüsseln abgelegt. Von dort aus kann der öffentliche Schlüssel später in eine eigene Datei kopiert werden, die Sie dann weitergeben können. Dann können Ihre Freunde und Bekannten Ihren öffentlichen Schlüssel zu ihrem Schlüsselbund hinzufügen.

Den privaten Schlüssel sollten Sie natürlich niemals weitergeben. Auch sollten Sie darauf achten, Ihr Schlüsselpaar selbst zu erzeugen und keine Schlüssel für Freunde zu erstellen♠ Die Sicherheit des gesamten Systems liegt darin, daß Ihr privater Schlüssel wirklich *privat* ist.

Der private Schlüssel sollte nach Möglichkeit nicht auf einem Rechner liegen, der für andere Benutzerinnen zugänglich ist, auch wenn er durch ein Mantra geschützt ist.

13.2.2. Einen Schlüssel in den Schlüsselbund aufnehmen

Um einen öffentlichen Schlüssel in den eigenen Bund mit öffentlichen Schlüsseln aufzunehmen, benutzen Sie folgenden Aufruf:

```
2.6.x: pgp -ka Datei [Schlüsselbund]
  5.0: pgpk -a Datei
  gpg: gpg --import Datei
```

♠ Vor ein paar Jahren sprachen die Techniker der Gesellschaft für Zahlungssysteme (gzs) noch davon, daß die SET-Schlüssel auf dem Rechner der Kunden erzeugt werden sollten. Inzwischen sieht die Realisierung leider so aus, daß die Kunden von ihrer Bank ein fertiges Zertifikat erhalten. SET ist ein Standard für sichere und gegenüber den Händlern anonyme Bezahlungen per Kreditkarte.

`Datei` ist diejenige Datei, die den oder die neuen Schlüssel enthält, der optionale Parameter `Schlüsselbund` bezeichnet die Datei, die den öffentlichen Schlüsselbund enthält.

Wird PGP mit diesem Befehl aufgerufen, dann prüft es zunächst, ob der Schlüssel schon bekannt ist. In diesem Fall wird er nicht ein weiteres Mal eingebunden, sondern auf neue Unterschriften und/oder Benutzer-IDs untersucht, die gegebenenfalls in den Schlüsselbund aufgenommen werden. PGP durchsucht die komplette Datei und bearbeitet alle darin enthaltenen Schlüssel auf diese Art und Weise. Was unterschriebene Schlüssel sind und wozu sie dienen, wird in Abschnitt 7.3 auf Seite 63 erläutert.

13.2.3. Einen Schlüssel oder eine Benutzer-ID löschen

Der Befehl hierfür lautet

```
 2.6.x: pgp -kr Benutzer-ID [Schlüsselbund]
   5.0: pgpk -r Benutzer-ID
   5.0: pgpk -ru Benutzer-ID
  gpg: gpg --delete-key Benutzer-ID
Menü: deluid
Menü: delkey
```

Wird ein Schlüssel gefunden, der zu der angegebenen ID paßt, fragt PGP, ob Sie diesen entfernen möchten, bzw. wenn der Schlüssel mehrere IDs hat, ob Sie ihn ganz entfernen möchten oder nur die eine oder andere ID. Wenn Sie aus Ihrem eigenen Schlüssel IDs löschen, hat das keine Auswirkungen auf Kopien Ihres Schlüssels, die bei anderen Leuten (oder auf Keyservern) diese IDs bereits gespeichert haben.

13.2.4. Inhaltsangabe des Schlüsselbunds

Diese erhalten Sie mit dem Befehl

```
 2.6.x: pgp -kv[v] [Benutzer-ID] [Schlüsselbund]
   5.0: pgpk -l[l] [Benutzer-ID]
  gpg: gpg -k[v[v]] [Benutzer-ID(s)]
```

Geben Sie eine Benutzer-ID an, werden alle Schlüssel aufgelistet, die den angegebenen Text enthalten, ansonsten alle Schlüssel der Datei (vgl.

Abb. 13.1). Geben Sie keinen Schlüsselbund an, wird `pubring.pgp` bzw. `pubring.pkr` bzw. `pubring.gpg` verwendet. Verwenden Sie die Option `-kvv`, werden zusätzlich zu den Schlüsseln alle Unterschriften ausgegeben. Bei PGP 5.0 bewirkt `-ll` statt `-l`, daß zu jedem Schlüssel auch sein Fingerprint ausgegeben wird.

```
[ccr@nescio ccr]$ pgp -kvv ccr
...
Schlüsselbund '/home/ccr/.pgp/pubring.pgp':
Suche nach Benutzer-ID "ccr":
Type Bits/KeyID    Date       User ID
pub  1024/B895FAD5 1997/06/05 Christopher Creutzig <ccr@math.uni-paderborn.de>
sig       6CE93239             Christopher Creutzig <christopher@nescio.zerberus.de>
sig       B895FAD5             Christopher Creutzig <ccr@math.uni-paderborn.de>
Es wurde ein passender Schlüssel gefunden.
[ccr@nescio ccr]$ gpg -kvv ccr
...
pub  1024D/32106275 1999-03-28 Christopher Creutzig <ccr@foebud.org>
sig        32106275 1999-03-28  Christopher Creutzig <ccr@foebud.org>
sub  1024g/678D1EF3 1999-03-28
sig        32106275 1999-03-28  Christopher Creutzig <ccr@foebud.org>
[ccr@nescio ccr]$ pgpk -ll ccr
Type Bits KeyID      Created    Expires    Algorithm       Use
pub  1024 0x32106275 1999-03-28 ---------- DSS             Sign & Encrypt
f20    Fingerprint20 = 8AFA B30A 453B 6BFD 6DDA  C3DA 2F65 4DBE 3210 6275
sub  1024 0x678D1EF3 1999-03-28 ---------- Diffie-Hellman
f20    Fingerprint20 = E856 AD21 3CAE A298 7082  A658 E56B 4A0A 678D 1EF3
sub  1024 0x06EF8875 1999-03-28 1999-04-11 Diffie-Hellman
f20    Fingerprint20 = 4536 B7E6 3B86 B0EB 162A  6786 A7A6 7097 06EF 8875
uid  Christopher Creutzig <ccr@foebud.org>
sig       0x32106275 1999-03-28 Christopher Creutzig <ccr@foebud.org>
```

Abbildung 13.1: `pgp -kvv ccr`

13.2.5. Einen Schlüssel in eine eigene Datei extrahieren

Um einen Schlüssel in eine eigene Datei zu kopieren (die Sie beispielsweise weitergeben können), verwenden Sie den Befehl

```
2.6.x: pgp -kx Benutzer-ID [Zieldatei] [Schlüsselbund]
  5.0: pgpk -x Benutzer-ID -o Zieldatei
  gpg: gpg --export Benutzer-ID > Zieldatei
  gpg: gpg --export Benutzer-ID | mail -s 'mein Key' a@b.de
```

Wird die Angabe der Zieldatei weggelassen, fragt PGP, wohin Sie den Schlüssel kopieren möchten. GnuPG gibt den Schlüssel auf die Standardausgabe aus, Sie können ihn also wie im Beispiel in eine Datei umleiten oder direkt in einem anderen Programm weiterverwenden.

Bei dieser Operation bleibt der Schlüsselbund vollständig, der Schlüssel wird nur kopiert.

Ist der Schlüssel unterschrieben, dann werden die Unterschriften ebenfalls mit kopiert. Wollen Sie den Schlüssel als Text verschicken (beispielsweise im UseNet, /CL-Netz, auf Ihrer Homepage oder auch als „Anhängsel“ an eine Nachricht), benutzen Sie den Befehl

```
2.6.x: pgp -kxa Benutzer-ID [Zieldatei] [Schlüsselbund]
  5.0: pgpk -xa Benutzer-ID -o Zieldatei
  gpg: gpg --export --armor Benutzer-ID >Zieldatei
```

13.2.6. Fingerabdruck anzeigen

Um einen Schlüssel über Telephon o. ä. vergleichen zu können, bietet sich die Verwendung des Fingerabdrucks an. Sie können ihn sich mit folgendem Befehl anzeigen lassen:

```
2.6.x: pgp -kvc Benutzer-ID
  5.0: pgpk -ll Benutzer-ID
  gpg: gpg --fingerprint Benutzer-ID
```

13.2.7. Einen Schlüssel unterschreiben

Mit dem folgenden Befehl wird ein Schlüssel unterschrieben:

```
2.6.x: pgp -ks Benutzer-ID [-u eigene_ID]
  5.0: pgpk -s Benutzer-ID
  gpg: gpg --sign-key [-u eigene_ID] Benutzer-ID
Menü: sign
```

Sie sollten eine solche Unterschrift nur leisten, wenn Sie sicher sind, daß Sie den Sinn und die Funktionsweise von Unterschriften unter Schlüsseln verstanden haben. Verstehen Sie diesen Warnhinweis bitte nicht als Arroganz von Leuten, die meinen, immer alles besser zu wissen. Wir haben in Diskussionen in verschiedenen Foren häufig feststellen müssen, daß die Frage, welche Voraussetzungen erfüllt sein müssen, damit man einen Schlüssel unterschreiben kann, das wahrscheinlich schwierigste Thema bei PGP ist. Auf keinen Fall sollten Sie mit Unterschriften unter PGP-Schlüsseln so großzügig sein, wie manche Prominente ihre Autogramme verteilen. Es geht hier nicht um Unterschriften im Sinne von

Autogrammen für Sammler, sondern um Unterschriften unter elektronische Dokumente. Näheres finden Sie in Abschnitt 7.1.2 ab Seite 60.

GnuPG bietet Ihnen mit dem Parameter `--lsign-key` auch die Möglichkeit, eine nicht exportierbare Signatur zu erstellen. Der Sinn und Zweck dieser Möglichkeit liegt darin, daß Sie evtl. den Schlüssel einer Zertifizierungsinstanz (deren Fingerprint Sie auf vertrauenswürdigem Weg bekommen haben oder den Sie direkt von dieser Instanz bekommen haben) lokal zu beglaubigen, ohne daß die Instanz mehrere Tausend Unterschriften unter ihrem Schlüssel sammelt. Bitte mißbrauchen Sie diese Möglichkeit nicht! Auch diese Sorte Unterschriften sollten Sie nur unter Schlüssel setzen, bei denen Sie zu 100% sicher sind, daß die angegebene ID korrekt ist.

Des weiteren bietet GnuPG die Möglichkeit, beim Unterschreiben eines Schlüssels mit `--set-policy-url URL` einen Verweis zu setzen, wo andere Menschen Hinweise darüber finden können, was Ihre Unterschrift Ihrer Meinung nach aussagt; darüber hinaus können Sie mit `--notation-data Typ=Wert` in die Unterschrift weitere Textdaten hineinsetzen, die integraler Bestandteil der Unterschrift werden. Beispielsweise könnte eine CA mit `--set-policy-url` einen Verweis auf ihren Haftungsausschluß und ähnliches in die Unterschrift mit aufnehmen und in Kombination mit `--notation-data` auch verschiedene Überprüfungsgrade anbieten, die alle mit demselben Schlüssel zertifiziert werden. Momentan können wir von letzterer Verwendungsart nur abraten: Es ist kaum anzunehmen, daß ein nennenswerter Teil der PGP-Nutzenden diese Daten überhaupt zur Kenntnis nimmt.

13.2.8. Einen Schlüssel zurückziehen

Nehmen wir an, Ihr Schlüssel ist (am Ende gar mit dem passenden Mantra) in fremde Hände gefallen. Das sollten Sie der ganzen Welt erzählen. Nicht, um bedauert zu werden, sondern damit niemand mehr diesem Schlüssel vertraut. Dafür müssen Sie eine Schlüssel-Rückrufurkunde ausstellen. Der Befehl hierfür lautet:

```
2.6.x: pgp -kd Ihre-ID
  5.0: pgpk --revoke Ihre-ID
  gpg: gpg --gen-revoke Ihre-ID
```

Diese Urkunde trägt dann auch noch Ihre Unterschrift und kann genau wie zuvor der öffentliche Schlüssel verschickt werden. Natürlich ist der

Sicherheitsverlust des Schlüssels nicht der einzig denkbare Grund dafür, ihn zurückzuziehen. In allen Fällen jedoch ist das Vorgehen dasselbe. Am besten schicken Sie Ihren neuen Schlüssel direkt mit.

Ach ja: Sobald Sie eine Rückrufurkunde ausgestellt haben, können Sie Ihren Schlüssel nicht mehr zum Unterschreiben verwenden, weil das keinen Sinn machen würde, die Unterschriften würden ohnehin nicht anerkannt werden.

13.2.9. Einen Schlüssel „abschalten“

Wenn Sie von jemandem überzeugt worden sind, daß sein geheimer Schlüssel abhanden gekommen ist, können Sie seinen bzw. ihren öffentlichen Schlüssel lokal mit dem Befehl

```
2.6.x: pgp -kd Benutzer-ID
  5.0: pgpk -d Benutzer-ID
```

„abschalten“. Hierbei wird allerdings keine Schlüssel-Rückrufurkunde ausgestellt und der Schlüssel kann mit demselben Befehl auch wieder angeschaltet werden. Ein abgeschalteter Schlüssel wird von PGP nicht für das Verschlüsseln verwendet, beim Überprüfen einer Unterschrift wird eine Warnung ausgegeben. GnuPG bietet diese Möglichkeit leider nicht.

13.3. Verschlüsseln einer Nachricht

Das Verschlüsseln einer Nachricht mit dem öffentlichen Schlüssel der Empfängerin geschieht mit folgendem Befehl:

```
2.6.x: pgp -e textdatei Benutzer-ID
  5.0: pgpe -r Benutzer-ID textdatei
 gpg: gpg -e -r Benutzer-ID textdatei
 gpg: gpg --encrypt --remote-user Benutzer-ID textdatei
```

Dieser Befehl erzeugt eine Datei namens `textdatei.pgp` bzw. `.gpg`, die den verschlüsselten Text enthält. Beispiel:

```
2.6.x: pgp -e brief.txt Alice
2.6.x: pgp -e brief.txt "Alice S"
  5.0: pgpe -r "Alice S" brief.txt
```

```
gpg: gpg -e -r Alice brief.txt
gpg: gpg -e -r "Alice S" brief.txt
```

Im ersten Beispiel durchsucht PGP den Bund mit den öffentlichen Schlüsseln nach einer ID, die das Wort „Alice“ enthält. Im zweiten Beispiel wird nach IDs gesucht, die „Alice S“ enthalten. Leerstellen in der ID-Angabe können nur benutzt werden, wenn die Angabe für die ID in Anführungszeichen eingeschlossen wird. Bei der Suche wird nicht zwischen Groß- und Kleinbuchstaben unterschieden.△ Wenn PGP eine passende ID findet, wird deren Schlüssel für das Chiffrieren der Datei `brief.txt` verwendet. Die Datei mit dem verschlüsselten Text heißt `brief.pgp`.

PGP versucht, die Datei mit dem Klartext zu komprimieren, bevor es sie verschlüsselt. Dies erhöht erheblich den Schutz gegen eine Kryptanalyse. Außerdem ist die verschlüsselte Datei in der Regel kleiner als die originale Klartextdatei.

Wenn die verschlüsselte Datei per E-Mail versandt werden soll, kann es sinnvoll sein, sie in druckbaren ASCII-Zeichen im Radix-64-Format darzustellen (siehe Abschnitt 13.10). Dies ist möglich durch Hinzufügen der Option `-a`. Moderne E-Mail-Systeme erlauben meistens aber direkt den Versand binärer Nachrichten.

13.4. Verschlüsseln einer Nachricht für mehrere Empfänger

Wenn dieselbe Nachricht an mehrere Empfängerinnen verschickt werden soll, kann sie so verschlüsselt werden, daß alle Empfängerinnen dieselbe verschlüsselte Nachricht entschlüsseln können. Beim Verschlüsseln kann man hierfür mehrere Benutzerinnen-IDs angeben. Beispiel:

```
2.6.x: pgp -e brief.txt Alice Bob Carol
  5.0: pgpe -r Alice -r Bob -r Carol brief.txt
gpg: gpg -e -r Alice -r Bob -r Carol brief.txt
```

Die durch dieses Kommando erzeugte Datei `brief.pgp` kann sowohl von Alice als auch von Bob oder Carol entschlüsselt werden. Es können belie-

△ Genaugenommen gibt es noch weitere Möglichkeiten, einen Schlüssel auszuwählen. Eine weitere ist die explizite Angabe der hexadezimalen Schlüssel-ID, GnuPG bietet auch die Möglichkeit, einen Schlüssel über seinen Fingerabdruck oder die genaue Angabe der gesamten Schlüssel-ID auszuwählen.

big viele Empfänger angegeben werden. Die Versionen 2.6.3i und 2.6.3in bieten auch die Möglichkeit, mit der Angabe `-@datei` den Inhalt der Datei `datei` als weitere Empfängerangaben zu lesen. Mit GnuPG läßt sich dieser Effekt erreichen, indem die Empfänger mit jeweils `remote-user` davor in eine Datei geschrieben werden und diese Datei dann mit Hilfe des Parameters `--options` eingelesen wird. (Nebenbei bemerkt läßt sich damit auch einstellen, daß alle Nachrichten auch an einen bestimmten weiteren Schlüssel kodiert werden sollen, beispielsweise den eigenen, womit ein Mail-Archiv auch für die verschlüsselten Nachrichten Sinn macht. Näheres zu dieser Überlegung finden Sie bei der Besprechung des Parameters `EncryptToSelf` bzw. `encrypt-to` auf Seite 123.)

13.5. Unterschreiben einer Nachricht

Der folgende Befehl unterschreibt eine Datei mit dem geheimen Schlüssel:

```
2.6.x: pgp -s textdatei [-u eigene-ID]
  5.0: pgps [-u eigene-ID] textdatei
  gpg: gpg -s [-u eigene-ID] textdatei
```

Der obige Befehl erzeugt eine Datei namens `textdatei.pgp` bzw. bei Verwendung von GnuPG `textdatei.gpg`.

Beispiel:

```
2.6.x: pgp -s brief.txt -u Bob
  5.0: pgps -u Bob brief.txt
```

Hier sucht PGP in dem Schlüsselbund mit den geheimen Schlüsseln nach einer ID, in der die Zeichenfolge „Bob“ vorkommt. Groß- und Kleinbuchstaben werden bei der Suche nicht unterschieden. Wenn PGP einen geheimen Schlüssel mit passender ID findet, wird dieser Schlüssel (nach Eingabe des korrekten Mantras) für die Unterschrift verwendet.

Wird `-u Benutzer-ID` nicht angegeben, dann verwendet PGP den default-Schlüssel für die Unterschrift. Näheres zur default-Einstellung finden Sie im Abschnitt `MYNAME` in Kapitel 14.

13.6. Unterschreiben und Verschlüsseln

Der folgende Befehl unterschreibt zuerst die Klartextdatei mit dem geheimen Schlüssel der Absenderin und verschlüsselt dann die Daten mit dem öffentlichen Schlüssel der Empfängerin:

```
2.6.x: pgp -es textdatei Empfängerin-ID [-u Benutzer-ID]
  5.0: pgpe -s -r Empfänger textdatei
 gpg: gpg -es -r Empfängerin-ID textdatei
```

Die mit dem obigen Befehl erzeugte Datei, die den unterschriebenen und anschließend verschlüsselten Text enthält, hat den Namen `textdatei.pgp`. Der für die Unterschrift verwendete geheime Schlüssel wird automatisch aus dem Bund mit geheimen Schlüssel herausgesucht. Der öffentliche Schlüssel der Empfängerin wird aus dem Bund mit öffentlichen Schlüsseln herausgesucht. Wenn in der Befehlszeile keine Empfänger-ID angegeben wird, fragt PGP/GnuPG nach.

Mehrere Empfängerinnen können durch einfaches Hinzufügen ihrer IDs in der Befehlszeile angegeben werden, genau wie beim einfachen Verschlüsseln.

13.7. Konventionelle Verschlüsselung

Manchmal kann es vorkommen, daß man eine Datei nach einer herkömmlichen Methode (mit einem geheimen Schlüssel, der nicht vom Programm zufällig gewählt wird) verschlüsseln möchte. Sinnvoll ist das beispielsweise für die Verschlüsselung einer Archivkopie von Daten, die einfach nur gespeichert, aber nicht an andere Leute verschickt werden soll. Der Befehl hierfür lautet:

```
2.6.x: pgp -c textdatei
  5.0: pgpe -c textdatei
 gpg: gpg -c textdatei
 gpg: gpg --symmetric [--cipher-algo algorithm] textdatei
```

Der Inhalt der Datei mit dem Namen `textdatei` wird komprimiert, verschlüsselt und in eine Datei mit dem Namen `textdatei.pgp` bzw. `textdatei.gpg` geschrieben. Bei dieser Operation werden keine öffentlichen Schlüssel, Schlüsselbunde, Benutzer-IDs usw. verwendet.

PGP fragt bei obigem Befehl nach einem Mantra, mit dem die Datei verschlüsselt werden soll. Dieses Mantra braucht nicht dasselbe zu sein, mit dem der geheime Schlüssel gesichert ist, und es sollte auch nicht dasselbe sein, da Sie es evtl. anderen Menschen mitteilen möchten. PGP versucht, die Daten vor der Verschlüsselung zu komprimieren. Die Algorithmen, die GnuPG anbietet, können Sie mit

```
gpg: gpg --version
```

abfragen. (Sie sind in der Zeile „Cipher:“ aufgelistet.)

13.8. Entschlüsselung und Prüfung der Unterschrift

Mit folgendem Befehl wird eine verschlüsselte Datei entschlüsselt und/oder eine Unterschrift geprüft:

```
2.6.x: pgp verschlüsselte_datei [-o klartext_datei]
  5.0: pgpv verschlüsselte_datei [-o klartext_datei]
  gpg: gpg [-o klartext_datei] verschlüsselte_datei
```

PGP geht davon aus, daß `verschlüsselte_datei` die Endung `.asc` oder `.pgp` trägt. Falls `verschlüsselte_datei` eine Unterschrift enthält, wird sie automatisch geprüft. Die Benutzer-ID der Unterschreibenden wird am Bildschirm angezeigt.

PGP bearbeitet `verschlüsselte_datei` vollautomatisiert. Die Datei kann sowohl nur verschlüsselt als auch nur unterschrieben oder auch unterschrieben und verschlüsselt sein. PGP verwendet die Schlüssel-ID, die in `verschlüsselte_datei` angegeben ist, um aus dem Bund mit den geheimen Schlüsseln denjenigen herauszusuchen, für den `verschlüsselte_datei` verschlüsselt wurde. Falls die Datei eine Unterschrift enthält, verwendet PGP die zusammen mit der Unterschrift gespeicherte Schlüssel-ID, um aus dem Bund mit öffentlichen Schlüsseln denjenigen herauszusuchen, mit dem die Unterschrift geprüft werden kann. Falls all diese Schlüssel vorhanden sind, ist während der Entschlüsselung keine weitere Eingabe erforderlich, ausgenommen natürlich das Mantra, mit dem der geheime Schlüssel gesichert ist.

Falls `verschlüsselte_datei` konventionell, also mit der Option `-c`, verschlüsselt wurde, fragt PGP bei der Entschlüsselung nach dem Mantra, mit dem die Datei verschlüsselt wurde.

Der optionale Parameter `-o klartext_datei` gibt an, in welcher Datei die entschlüsselten Daten gespeichert werden sollen. Fehlt diese Angabe, wird bei PGP 2.6.x der Name von `verschlüsselte_datei` ohne Suffix für die Klartextdatei verwendet, mit Ausnahme der Version 2.6.3i, die die Endung nach dem Dateiinhalt vergibt – bedauerlicherweise auch entgegen der Vorschrift des Users durch `-o`. GnuPG verwendet standardmäßig den Originaldateinamen, der in der verschlüsselten Datei gespeichert ist. Wird der Befehl

gpg: `gpg --decrypt dateiname`

verwendet, so schreibt GnuPG den entschlüsselten Text auf die Standardausgabe. Sie können ihn mit

gpg: `gpg --decrypt datei1 >datei2`

in eine Datei `datei2` umleiten.

13.9. Erzeugen einer Datei mit Zufallszahlen

Seit der Version 2.6.2 bietet PGP die Möglichkeit, den internen Generator für Pseudozufallszahlen auch dazu zu verwenden, eine Datei mit kryptographisch zuverlässigen Zufallszahlen zu erstellen. Dazu dient der Aufruf

2.6.x: `pgp +makerandom=Bytes Dateiname`
gpg: `gpg --gen-random Qualität [Bytes] >Dateiname`

der eine Datei `Dateiname` der Größe `Bytes` mit pseudozufälligem Inhalt erzeugt. Bei GnuPG haben Sie darüber hinaus die Möglichkeit, mit der Angabe `0`, `1` oder `2` verschieden „stark zufällige“ Zahlen zu erzeugen. PGP 5.0 scheint keine derartige Option zu bieten.

13.10. Nachrichten im Radix-64-Format

In vielen E-Mail-Systemen ist nur der Versand von ASCII-Text möglich. Meistens werden die nötigen Konvertierungen, um Texte mit Umlauten oder Binärdaten zu versenden, bereits vom E-Mail-System vorgenommen, aber das funktioniert nicht immer und ist manchmal überhaupt nicht vorgesehen. Binärdaten, wie die von PGP normalerweise erzeugten

verschlüsselten Dateien, können dann nicht versandt werden. PGP kann deshalb bei Bedarf die verschlüsselten Daten im Radix-64-Format darstellen, ähnlich dem Privacy-Enhanced-Mail-Format oder der MIME-Kodierung desselben Namens.

Radix-64 stellt binäre Daten ausschließlich durch druckbare 6-Bit-ASCII-Zeichen dar, so daß eine verschlüsselte Nachricht wie gewöhnlicher E-Mail-Text verschickt werden kann. Radix-64 ist also eine Art „Transport-Verpackung“, die Schutz vor einer „Verstümmelung“, also einer Veränderung, die das Entschlüsseln der Nachricht unmöglich machen würde, auf dem Transportweg bietet. Um Übertragungsfehler erkennen zu können, wird der Radix-64-Darstellung eine Prüfsumme hinzugefügt.

Im Radix-64-Format werden jeweils drei Byte durch vier druckbare ASCII-Zeichen dargestellt, so daß die Länge der Datei um etwa 33 Prozent zunimmt. Das sieht zunächst nach einer ziemlich großen Aufblähung der Datei aus, berücksichtigt werden muß aber, daß PGP Klartextdateien häufig um einen größeren Faktor komprimiert, bevor sie verschlüsselt werden.

Für eine Radix-64-Darstellung der verschlüsselten Datei wird einfach die Option `-a` beim Programmaufruf hinzugefügt:

```
2.6.x: pgp -esa brief.txt Empfänger
  5.0: pgpe -sa -r Empfänger brief.txt
  gpg: gpg -esa -r Empfänger brief.txt
```

Hier wird `brief.txt` unterschrieben und komprimiert, anschließend verschlüsselt und das Ergebnis im Radix-64-Format in eine Datei mit dem Namen `brief.txt.asc` (bzw. unter MS-DOS `brief.asc`) geschrieben. Diese Datei kann wie gewöhnliche E-Mail im Internet oder jedem anderen E-Mail-Netzwerk verschickt werden. Die Entschlüsselung einer so verschickten Nachricht unterscheidet sich nicht von der Entschlüsselung einer `.pgp`-Datei:

```
2.6.x: pgp brief
  5.0: pgpv brief
  gpg: gpg brief.asc
```

PGP sucht hier zuerst nach einer Datei namens `brief.asc` und erst danach nach `brief.pgp`. PGP erkennt automatisch am Dateiinhalt, daß `brief.asc` vor der eigentlichen Entschlüsselung erst wieder zurück in Binärdarstellung umgewandelt werden muß.

In manchen E-Mail-Netzen ist der Versand sehr langer Nachrichten nicht möglich.⊖ Längere Texte müssen dort in mehrere Teile gesplittet werden, die einzeln verschickt werden. Wenn beim Verschlüsseln die Option für Darstellung im Radix-64 Format angegeben wurde, schreibt PGP bei einem langen Text die verschlüsselten Daten in mehrere Dateien, deren Namen auf `.as1`, `.as2`, `.as3` usw. enden. Das PGP der Empfängerin fügt diese automatisch wieder zusammen, wenn beim Aufruf von PGP einfach die erste Datei angegeben wird. Bei der Entschlüsselung ignoriert PGP allen Text aus den Nachrichtenköpfen, der nicht zu den Radix-64-Blöcken gehört. Die Größe dieser Dateien läßt sich über den Parameter `ARMORLINES` einstellen, vgl. Abschnitt 14 auf Seite 113.

Möchte man einen öffentlichen Schlüssel im Radix-64-Format verschicken, kann die Option `-a` bzw. `--armor` auch beim Befehl für das Extrahieren des Schlüssels aus der Datei mit öffentlichen Schlüsseln angegeben werden.

Hat man vergessen, die Option `-a` beim Verschlüsseln einer Nachricht oder beim Extrahieren eines Schlüssels anzugeben, kann man die Binärdatei auch nachträglich in das Radix-64-Format umwandeln, indem man PGP nur mit der Option `-a` aufruft, ohne Angabe der Option für Verschlüsseln oder Unterschreiben:

2.6.x: `pgp -a brief.pgp`

Um keine Mißverständnisse aufkommen zu lassen: Dieses „Nacharbeiten“ muß mit der bereits verschlüsselten Datei erfolgen, also mit `brief.pgp` in obigem Beispiel. *Falsch* wäre die folgende Befehlskombination:

2.6.x: `pgp -es brief.txt Benutzer-ID`
2.6.x: `pgp -a brief.txt`

Der erste Befehl erzeugt eine verschlüsselte Datei `brief.pgp`; der zweite Befehl erzeugt eine Datei `brief.asc` im Radix-64-Format, jedoch aus der Klartextdatei `brief.txt`. Daß `brief.asc` nicht unmittelbar „für das menschliche Auge lesbar“ ist, bedeutet *nicht,* daß die Datei verschlüsselt ist!

PGP 5.0 scheint keine solche Funktion anzubieten; um den Effekt mit GnuPG zu erreichen, brauchen Sie den Aufruf

gpg: `gpgm --enarmor datei.gpg`

⊖ Beispielsweise ist nicht garantiert, daß eine Internet-E-Mail mit 1MB Länge beim Empfänger ankommt

13.11. Klartext-Unterschrift

Wenn man eine Datei versenden möchte, die zwar unterschrieben, aber nicht verschlüsselt ist, komprimiert PGP normalerweise alle Daten und wandelt sie anschließend gegebenenfalls in das Radix-64-Format. Handelt es sich bei den Daten um einen Text, ist er folglich nicht unmittelbar lesbar.

Soll der Text lesbar bleiben, bietet PGP die Möglichkeit, nur die Unterschrift im Radix-64-Format an den Text anzufügen. Empfängerinnen einer solcher Nachricht brauchen also PGP nicht aufzurufen, um den Text zu lesen. PGP ist hier nur für eine Kontrolle der Unterschrift erforderlich. Der entsprechende Befehl lautet

```
2.6.x: pgp -sat +clearsig=on presseerklaerung.txt
  5.0: pgps -at --clearsig=on presseerklaerung.txt
  gpg: gpg --clearsig presseerklaerung.txt
```

Das Ergebnis sieht dann z. B. so aus:

```
-----BEGIN PGP SIGNED MESSAGE-----


Wir verwenden ab sofort PGP für unsere Nachrichten.


-----BEGIN PGP SIGNATURE-----
Version: 2.6.3ia
Charset: noconv

iQCVAgUBNvUMgRmebHZs6TI5AQHc6QP+NOmcXeefCrvcJdgaAk9d7LQuZFAgL8Vc
QORHr3O4f6A8kGZ5vCALGx892dgaZs29EhBtE2yLKewgdRB3BCmpFmwemDORH8Zw
C2OuVPuOWx7rM+uy410BdJDI5GQFstXVzmvD/yxL+ZgLhBFSDI1sTMhFv8RWahx5
TnKbgD1e8OM=
=OTjk
-----END PGP SIGNATURE-----
```

Wichtiger Hinweis: Eine solche Nachricht wird als Textnachricht versandt und unterliegt dadurch möglichen Änderungen auf dem Versandweg. Typische Änderungen sind Zeichensatzkonvertierungen oder das Einfügen oder Löschen von Leerzeichen (primär am Zeilenende). PGP toleriert gewisse Änderungen, beispielsweise werden Leerzeichen am Zeilenende nicht berücksichtigt und eine Zeichensatzkonvertierung vom Standardzeichensatz des Absenders zum Standardzeichensatz der Empfängerin wird auch toleriert – ein anderes Vorgehen würde die

Klartextunterschriften von vornherein unbrauchbar machen. Wenn die Änderungen anderer Natur sind, wird PGP die Nachricht als verändert erkennen, was zu einem in diesem Fall unberechtigten Verdacht einer echten, inhaltlichen Fälschung führt. Dieses Problem haben alle Klartext-Signaturverfahren aus Prinzip, trotzdem schien es sinnvoll, diese Möglichkeit des unterschriebenen Klartextes trotz ihrer Störanfälligkeit in PGP aufzunehmen. In der Realität hat sich gezeigt, daß das Verfahren erstaunlich stabil arbeitet. Ein weiterer Störfaktor kommt durch die Neuerungen in PGP 2.6.3i hinzu: Hier wird in der Klartextunterschrift der verwendete Zeichensatz mit abgelegt. So eine Nachricht kann natürlich nur dann als korrekt unterschrieben angesehen werden, wenn sie entweder keine Umlaute und sonstigen Sonderzeichen enthält oder der Empfänger die Nachricht zufälligerweise im selben Zeichensatz erhält, wie der Absender sie geschrieben hat.

In der CLEARSIG-Routine vor den Versionen 2.6.i und 2.6.2 (siehe Anhang A auf Seite 264: „Die vielen PGP-Versionen“) gab es einen Konzeptfehler, der es erlaubte, Nachrichten in gewisser Weise zu verändern, ohne daß dies von PGP gemeldet wurde. Die mit diesem Buch ausgelieferte Version 2.6.2i hat diesen Fehler nicht mehr.

Falls die Details Sie interessieren: Wenn PGP eine Klartext-Unterschrift überprüfen wollte, wurde Text, der nach der Zeile

```
-----BEGIN PGP SIGNATURE-----
```

folgte, nicht beachtet, bis eine Leerzeile gefunden wurde. Grund dafür war, daß hier dieselbe Routine wie für das Einlesen der Unterschrift verwendet wurde. Dort muß die Zeile

```
Version x.y
```

übersprungen werden. Da es auch möglich war, Zeilen, die PGP nicht als Leerzeile ansah, aber für den Anwender wie Leerzeilen aussehen, einzufügen, ließen sich Klartext-Unterschriften bei flüchtiger Kontrolle fälschen.

13.12. Die Umgebungsvariable für das PGP-Verzeichnis: PGPPATH

PGP benötigt beim Ver- und Entschlüsseln mehrere Dateien, unter anderem die beiden Dateien pubring.pgp und secring.pgp bzw. bei PGP 5.0

`pubring.pkr` und `secring.skr` mit öffentlichen und geheimen Schlüsseln, `randseed.bin` (enthält die Parameter für den Zufallszahlengenerator), `config.txt` bzw. `pgp.cfg` (Konfigurationsdatei) und `language.txt` bzw. `language50.txt` (enthält die Textmeldungen von PGP, unter Umständen in mehreren Sprachen).

Diese Dateien können (und sollten) in einem eigens dafür angelegten Verzeichnis stehen, z. B. `c:\pgp` oder `~/.pgp/`. Damit PGP diese Dateien auch dann findet, wenn es aus einem beliebigen anderen Verzeichnis aufgerufen wird, muß die Umgebungsvariable `PGPPATH` auf das Verzeichnis mit den PGP-Dateien gesetzt werden. Unter MS-DOS geschieht das mit dem Befehl

```
SET PGPPATH=C:\PGP
```

Wenn `PGPPATH` so gesetzt ist, benutzt PGP als Datei mit den öffentlichen Schlüsseln die Datei `c:\pgp\pubring.pgp` (vorausgesetzt, Verzeichnis und Datei existieren). Mit einem geeigneten Editor♣ kann unter MS-DOS der Befehl `SET PGPPATH=` in die Datei `autoexec.bat` eingetragen werden, so daß `PGPPATH` automatisch beim Start des Rechners gesetzt wird. Unter Unix kann sich jeder User in seiner `~/.profile`, `~/.cshrc` oder `~/.bashrc` eine entsprechende Zeile einfügen, für `sh`-ähnliche Shells also

```
PGPPATH=~/.pgp/
export PGPPATH
```

bzw. für `csh`-ähnliche Shells

```
setenv PGPPATH ~/.pgp/
```

Wenn `PGPPATH` nicht definiert ist, sucht PGP die Dateien im aktuellen Verzeichnis. Unter Unix werden einige Standardverzeichnisse durchsucht, u. a. `~/.pgp`, o. g. Einstellung ist also überflüssig. Wenn Ihre PGP-Version ihre Daten nicht findet, teilt sie Ihnen mit, wo sie sucht. Die entsprechende Umgebungsvariable für GnuPG nennt sich `GNUPGHOME`, standardmäßig wird das Verzeichnis `~/.gnupg` verwendet.

♣ D. h. insbesondere keine Textverarbeitung wie z. B. Word. Für Konfigurationsdateien und E-Mails sind Textverarbeitungen nicht empfehlenswert.

13.13. Kurzübersicht über die PGP-Befehle

13.13.1. Kommandos zum Umgang mit Nachrichten

- Zum Verschlüsseln eines Klartextes mit dem öffentlichen Schlüssel der Empfängerin:

```
2.6.x: pgp -e datei Empfängerin-ID
 5.0 : pgpe -r Empfängerin-ID datei
 gpg : gpg -e -r Empfängerin-ID datei
 gpg : gpg --encrypt --remote-user Empfängerin-ID datei
```

- Zum Unterschreiben eines Klartextes mit dem geheimen Schlüssel:

```
2.6.x: pgp -s textdatei [-u eigene_ID]
 5.0 : pgps [-u eigene_ID] textdatei
 gpg : gpg -s [-u eigene_ID] textdatei
 gpg : gpg --sign [--local-user eigene_ID] textdatei
```

- Zum Unterschreiben eines Klartextes mit dem geheimen Schlüssel und anschließendem Verschlüsseln des Klartextes mit dem öffentlichen Schlüssel des Empfängers:

```
2.6.x: pgp -es textdatei Empfänger-ID [-u eigene_ID]
 5.0 : pgpe -r Empfänger-ID -s [-u eigene_ID] textdatei
 gpg : gpg -es -r Empfänger-ID [-u eigene_ID] textdatei
 gpg : gpg --encrypt --remote-user Empfängerin-ID \
         --sign [--local-user eigene_ID] textdatei
```

- Zum Verschlüsseln eines Klartextes nur mit herkömmlicher Verschlüsselung (kein asymmetrisches Verfahren, sondern gleicher Schlüssel zum Ver- und Entschlüsseln):

```
2.6.x: pgp -c textdatei
 5.0 : pgpe -c textdatei
 gpg : gpg -c textdatei
 gpg : gpg --symmetric textdatei
```

- Zum Entschlüsseln einer verschlüsselten Datei, oder um die Echtheit einer Unterschrift einer unterschriebenen Datei zu prüfen:

```
2.6.x: pgp verschlüsselte_datei [-o Klartextdatei]
 5.0 : pgpv verschlüsselte_datei [-o Klartextdatei]
 gpg : gpg verschlüsselte_datei [-o Klartextdatei]
 gpg : gpg --decrypt verschlüsselte_datei \
           [--output Klartextdatei]
 gpg : gpg --verify signierte_datei
```

- Um eine Nachricht für beliebig viele Empfänger zu verschlüsseln:

```
2.6.x: pgp -e textdatei Benutzer-ID1 Benutzer-ID2
 5.0 : pgpe -r ID1 -r ID2 textdatei
 gpg : gpg -e -r ID1 -r ID2 textdatei
```

13.13.2. Kommandos zur Schlüsselverwaltung

- Um ein einzigartiges persönliches Paar aus öffentlichem und privatem Schlüssel herzustellen:×

```
2.6.x: pgp -kg
 5.0 : pgpk -g
 gpg : gpg --gen-key
```

- Zum Hinzufügen eines öffentlichen oder geheimen Schlüssels in den öffentlichen oder geheimen Schlüsselbund:

```
2.6.x: pgp -ka Schlüsseldatei [Schlüsselbund]
 5.0 : pgpk -a Schlüsseldatei
 gpg : gpg --import Schlüsseldatei \
           [--keyring Schlüsselbund]
```

- Zum Extrahieren eines Schlüssels aus dem öffentlichen oder geheimen Schlüsselbund:

```
2.6.x: pgp -kx Benutzer-ID Schlüsseldatei
2.6.x: pgp -kxa Benutzer-ID Schlüsseldatei
 5.0 : pgpk -x Benutzer-ID -o Schlüsseldatei
 5.0 : pgpk -xa Benutzer-ID -o Schlüsseldatei
 gpg : gpg --export Benutzer-ID >Schlüsseldatei
 gpg : gpg --export --armor Benutzer-ID >Schlüsseldatei
```

× In diesem Zusammenhang *könnte* der Abschnitt NOMANUAL des Kapitels 14 interessant sein, falls Sie Probleme haben.

- Zum Anzeigen des Inhalts des öffentlichen Schlüsselbunds:

```
2.6.x: pgp -kv[v] [Benutzer-ID] [Schlüsselbund]
 5.0 : pgpk -l[l] [Benutzer-ID]
 gpg: gpg -k[v[v]] [Benutzer-ID] [Schlüsselbund]
 gpg: gpg --list-keys [Benutzer-ID(s)]
 gpg: gpg --list-sigs [Benutzer-ID(s)]
```

- Zum Anzeigen des „Fingerabdrucks“ eines öffentlichen Schlüssels, um ihn (z. B. über Telephon) mit seinem Besitzer vergleichen zu können:

```
2.6.x: pgp -kvc [Benutzer-ID] [Schlüsselbund]
 5.0 : pgpk -ll [Benutzer-ID]
 gpg: gpg -k[v]c [Benutzer-ID] [Schlüsselbund]
 gpg: gpg --fingerprint [Benutzer-IDs]
```

- Zum Anzeigen des Inhalts und zur Überprüfung der Unterschriften des öffentlichen Schlüsselbunds:

```
2.6.x: pgp -kc [Benutzer-ID] [Schlüsselbund]
 5.0 : pgpk -c [Benutzer-ID]
 gpg: gpg --check-sigs [Benutzer-IDs]
```

- Zum Bearbeiten der Vertrauensparameter eines fremden öffentlichen Schlüssels:

```
2.6.x: pgp -ke Benutzer-ID [Schlüsselbund]
 5.0 : pgpk -e Benutzer-ID
 gpg: gpg --edit-key Benutzer-ID
```

- Zum Entfernen eines Schlüssels oder nur einer Benutzer-ID aus einem öffentlichen Schlüsselbund:

```
2.6.x: pgp -kr [Benutzer-ID] [Schlüsselbund]
 5.0 : pgpk -r [Benutzer-ID]
 5.0 : pgpk -ru [Benutzer-ID]
 gpg: gpg --delete-key Benutzer-ID
 gpg: gpg --delete-secret-key Benutzer-ID
```

- Zum Unterschreiben und Beglaubigen eines öffentlichen Schlüssels im öffentlichen Schlüsselbund:

```
2.6.x: pgp -ks [Benutzer-ID] [-u eigene_Benutzer-ID] \
           [Schlüsselbund]
  5.0: pgpk -s Benutzer-ID [-u eigene_Benutzer-ID]
  gpg: gpg --sign-key Benutzer-ID [-u eigene_Benutzer-ID]
```

- Zum Entfernen ausgewählter Unterschriften einer Benutzer-ID eines Schlüsselbunds[◇]:

```
2.6.x: pgp -krs [Benutzer-ID] [Schlüsselbund]
  5.0: pgpk -rs Benutzer-ID
  gpg: gpg --edit-key Benutzer-ID
Menü: delsig
```

- Zum dauerhaften Widerrufen des eigenen Schlüssels durch Erzeugen einer Schlüssel-Rückrufurkunde:

```
2.6.x: pgp -kd eigene_ID
  5.0: pgpk --revoke eigene_ID
  gpg: gpg --gen-revoke eigene_ID
```

- Zum dauerhaften Widerrufen einer eigenen Unterschrift unter dem Schlüssel einer anderen Person:[⊗]

```
  5.0: pgpk --revokes Benutzer_ID
  gpg: gpg --edit-key Benutzer_ID
Menü: revsig
```

- Zum Sperren oder Entsperren eines öffentlichen Schlüssels im eigenen öffentlichen Schlüsselbund:

```
2.6.x: pgp -kd Benutzer-ID
  5.0: pgpk -d Benutzer-ID
```

◇ Die Bedienung bei GnuPG ist etwas verworren und anscheinend nur hier erklärt: Sie müssen zunächst eine der Benutzer-IDs des Schlüssels anwählen, indem Sie die Nummer als Kommando eingeben, also beispielsweise einfach `1`. Als nächsten Schritt können Sie dann `delsig` aufrufen und werden dann für jede Unterschrift gefragt, ob Sie sie löschen möchten.

⊗ Diese Möglichkeit sollte Sie nicht dazu verleiten, unvorsichtig mit Unterschriften umzugehen!

13.13.3. Selten verwendete Kommandos

- Zum Entschlüsseln einer Nachricht, wobei die Unterschrift intakt bleibt:

2.6.x: `pgp -d verschlüsselte_Datei`

- Zum Erstellen einer Unterschriftsbescheinigung, die vom unterschriebenen Dokument getrennt ist:

2.6.x: `pgp -sb textdatei [-u eigene_Benutzer-ID]`
gpg: `gpg -[s]b textdatei`
gpg: `gpg --detach-sign textdatei`

- Zum Trennen einer Unterschriftsbescheinigung vom unterschriebenen Dokument:

2.6.x: `pgp -b verschlüsselte_Datei`

- Zum Erzeugen eines Schlüssels unter direkter Angabe der Parameter auf der Kommandozeile:

```
2.6.x: pgp -kg [Schlüssellänge [Länge des Exponenten]]
 5.0 : pgpk -g RSA|DSS Größe Username Ablaufdatum Mantra
 5.0 : pgpk +fastkeygen=0 -g RSA 2048 \
          'Christopher Creutzig <ccr@foebud.org>' \
          never 'mein kleines, süßes Mantra...'
```

Bei der Version 2.6.3in können Sie noch zusätzlich eine bestimmte ID gezielt erzeugen. Es ist davon abzuraten, mehrere Schlüssel mit derselben ID zu erzeugen, da dies die Schlüsselverwaltung der meisten anderen Versionen komplett durcheinanderbringt.

13.13.4. Kommandooptionen, die in Verbindung mit anderen Optionen benutzt werden können

- Um die erzeugte Ausgabedatei nicht im Binärformat zu erhalten, sondern im Radix-64-Format, also nur aus druckbaren Zeichen bestehend, können Sie einfach den Schalter `-a` anhängen, wenn ein Dokument verschlüsselt oder unterschrieben wird, oder wenn ein Schlüssel entnommen wird. Für GnuPG lautet die entsprechende Option `--armor`. Näheres zu Radix-64 finden Sie in Abschnitt 13.10.

```
2.6.x: pgp -sea textdatei Benutzer_ID
2.6.x: pgp -kxa Benutzer-ID Schlüsseldatei \
            [Schlüsselbund]
 5.0 : pgpk -xa Benutzer-ID -o Schlüsseldatei
 5.0 : pgpe -sa -r Benutzer-ID textdatei
 gpg: gpg --sign --armor programm.exe
```

- Zum physikalischen Löschen der Klartextdatei nach der Erstellung der verschlüsselten Datei die Option `-w` anfügen:

```
2.6.x: pgp -sew brief.txt EmpfängerID
```

- Um festzulegen, daß eine Klartextdatei Text und keine Binärdaten enthält und beim Empfänger in den richtigen Zeichensatz gewandelt wird, können Sie die `-t` Option zu den anderen Optionen hinzufügen. GnuPG bietet alternativ auch `--textmode`. Sowohl PGP als auch GnuPG erkennen Textdateien für gewöhnlich ohne Probleme, so daß diese Option weniger wichtig ist, als es vielleicht scheint.

```
2.6.x: pgp -seat brief.txt EmpfängerID
 5.0 : pgpe -sat -r EmpfängerID brief.txt
 gpg: gpg -set brief.txt -r EmpfängerID
```

- Zum Anzeigen des entschlüsselten Klartextes auf dem Monitor, ohne ihn in eine Datei zu schreiben, können Sie das Kommando `-m` benutzen:

```
2.6.x: pgp -m verschlüsselte_Datei
 5.0 : pgpv -m verschlüsselte_Datei
```

- Um festzulegen, daß der entschlüsselte Klartext der Empfängerin nur auf ihrem Bildschirm angezeigt wird und nicht auf Diskette/Festplatte gesichert werden soll, die Option `-m` anhängen:

```
2.6.x: pgp -steam brief.txt EmpfängerID
```

- Zum Wiederherstellen des Original-Dateinamens beim Entschlüsseln die Option `-p` anfügen:

```
2.6.x: pgp -p verschlüsselte_Datei
```

- Um den Filterbetrieb im Unix-Stil zu benutzen (Lesen von Standardeingabe und Schreiben auf die Standardausgabe), benutzen Sie die Option `-f`.☉ Bei GnuPG ist dieses Verhalten Standard, wenn keine Dateinamen angeben werden.

2.6.x:
```
ls -al | pgp -feast Empfänger-ID \
       | mail -s '** Kein Betreff **' emp@faenger.de
```
5.0:
```
ps ax | pgpe -fast -r Empfänger \
       | mail -s '** Kein Betreff **' emp@faenger.de
```
gpg:
```
cat signed-file | gpg | wc -l
```
gpg:
```
echo "hallo" | gpg -ear tom \
       | mail t.budewig@bionic.zerberus.de
```

☉ Die Angabe `'** Kein Betreff **'` ist hier durchaus als gutes Beispiel anzusehen: Es ist inkonsequent, eine Nachricht zu verschlüsseln und der gesamten lauschenden Welt trotzdem mitzuteilen, was drinsteht.

14. Konfigurierbare Parameter: config.txt/pgp.cfg/options

Die Datei `config.txt` (PGP 2.6x) bzw. `pgp.cfg` (PGP 5.0) enthält eine Reihe von Parametern, mit denen PGP den individuellen Bedürfnissen angepaßt werden kann. Die `config.txt` steht im PGP-Verzeichnis. Das Analogon für GnuPG ist die Datei `options` im GnuPG-Verzeichnis. In dieser Datei stehen Optionen, eventuell gefolgt von Parametern, die auch (mit zwei Strichen vorangestellt) auf der Kommandozeile stehen können. Im Folgenden finden Sie daher bei den Angaben der Einträge der `config.txt` die entsprechenden Kommandozeilenparameter für GnuPG.

GnuPG kennt einige weitere Optionen, die aber recht speziell sind und hier nicht besprochen werden. Sie finden sie aber alle via

```
man gpg
```

Ebenso hat PGP 5.0 einige Einstellungsmöglichkeiten, auf die wir in diesem Kapitel nicht näher eingehen:

- Zur Erläuterung der Option `FastKeyGen` möchten wir Sie auf Kapitel 17.1 ab Seite 146 verweisen.
- Die Optionen `AutoServerFetch`, `HTTPKeyServerHost` und `HTTPKeyServerPort` betreffen die Kommunikation mit einem Keyserver (was nur für Anwender mit Standleitung interessant ist).
- Die Option `RandomDevice` können Sie verwenden, um eine andere Quelle als `/dev/random` für gute Zufallszahlen anzugeben – wenn dieser Wert nicht gesetzt ist oder das angegebene Device nicht gefunden wird, bezieht PGP 5.0 seine Zufallszahlen genau wie PGP 2.6.x aus Tastatureingaben.
- Die Optionen `WarnOnMixRSADiffieHellman` und `WarnOnRSARecipAndNonRSASigner` sollten Sie eingeschaltet lassen; PGP warnt Sie

dann, wenn Sie Nachrichten verschlüsseln bzw. Unterschriften leisten, die die Empfänger unter Umständen (mit PGP 2.6.x) nicht verwenden können.

Die US-englischen Erläuterungen bekommen Sie (auf Unix-Systemen) mit dem folgenden Befehl angezeigt:

```
man pgp.cfg
```

In `config.txt` kann beispielsweise eingestellt werden, in welchem Verzeichnis PGP temporäre Dateien speichert, in welcher Sprache PGP seine Meldungen ausgibt, oder wie skeptisch sich PGP bei der Prüfung von Unterschriften unter öffentlichen Schlüsseln verhält. Die einzelnen Konfigurationsparameter können je nach Typ als Wert ganze Zahlen, Zeichenketten (also Text), oder „on/off“ haben. Eine Beispielkonfiguration, an der man sich bei der individuellen Einstellung orientieren kann, ist PGP beigelegt.

Leere Zeilen werden in `config.txt` ignoriert, ebenso alles, was in einer Zeile rechts von einem #, der Kommentarmarkierung, steht. Bei den Parameternamen wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Beachten Sie, daß in der Beispielkonfiguration die Zeilen für die Einstellung mancher Parameter ebenfalls mit einem # beginnen. Für die Aktivierung dieser Parametereinstellungen muß das # am Zeilenanfang gelöscht werden. Die Verwendung eines # ist auch sinnvoll, um mehrere Parametereinstellungen auszuprobieren, ohne die Texte der einzelnen Einstellungen zu löschen.

Beispiel:

```
# Die folgende Einstellung ist besser, wenn Texte
# zu ver- oder entschlüsseln sind, die
# unter Windows bearbeitet wurden:
# charset = latin1
# Für MS-DOS ist das folgende besser:
charset = cp850
```

Hier wertet PGP nur die Zeile `charset = cp850` aus; die auskommentierte Zeile `charset = latin1` wird ignoriert. Die obere Einstellung kann durch einfaches Umstellen des # aktiviert werden.

Ein Ausschnitt aus einer typischen Konfigurationsdatei:

```
# TMP is the directory for PGP scratch files,
# such as a RAM disk.
# Can be overridden by environment variable TMP.
TMP = "e:\temp\"
# Use -a flag
# for ASCII armor whenever applicable.
Armor = on
# CERT_DEPTH is how deeply
# introducers may introduce introducers.
cert_depth = 3
```

Wenn bestimmte Parameter nicht in `config.txt` definiert sind oder wenn diese Datei nicht existiert bzw. PGP die Datei nicht findet, setzt es automatisch sinnvolle Standardwerte ein.

Die Parameter aus `config.txt` können auch in der Kommandozeile angegeben werden; bei PGP 2.6.x muß hierfür ein + vorangestellt werden, PGP 5.0 erwartet `--`. Dadurch ist es möglich, im Einzelfall mit anderen Parametern zu arbeiten, ohne daß `config.txt` extra hierfür geändert werden muß. Die beiden Kommandos im nachfolgenden Beispiel liefern dasselbe Ergebnis:

2.6.x: `pgp -e +armor=on brief.txt mueller`
2.6.x: `pgp -ea brief.txt mueller`

`TMP` – Name des Verzeichnisses für temporäre Dateien

Standardeinstellung: `TMP = ""`

`TMP` gibt an, welches Verzeichnis PGP für temporäre Dateien verwendet. Ein sinnvoller Platz für temporäre Dateien ist – falls vorhanden – eine RAM-Disk (also ein virtuelles Laufwerk im Hauptspeicher Ihres Rechners). Bei Verwendung einer RAM-Disk wird PGP etwas schneller, zudem wird die Sicherheit ein wenig gesteigert. Wenn `TMP` nicht definiert ist, werden temporäre Dateien im aktuellen Verzeichnis gespeichert. Falls eine Umgebungsvariable `TMP` definiert ist, verwendet PGP deren Wert als Namen des Verzeichnisses für temporäre Dateien. GnuPG verwendet keine temporären Dateien.

LANGUAGE – Auswahl der Sprache für Textmeldungen von PGP

Standardeinstellung: LANGUAGE = en

(GnuPG wertet die Umgebungsvariable LANG aus.) PGP gibt eine Reihe von Anfragen, Warnungen und Erläuterungen am Bildschirm aus. Normalerweise erscheinen diese Texte auf US-englisch. PGP kann so angepaßt werden, daß es diese Meldungen in anderen Sprachen ausgibt, ohne daß die Datei mit dem ausführbaren Programm geändert werden muß.

Eine Reihe von Menschen aus verschiedenen Ländern haben die Meldungen von PGP in ihre Muttersprache übersetzt. Diese übersetzten Texte stehen in einer speziellen Datei namens language.txt bzw. language50.txt, die im PGP-Programmpaket enthalten ist. Die Datei language.txt kann die Meldungen in mehreren Sprachen enthalten. Zur Zeit existieren neben den originalen englischen Texten Übersetzungen in Deutsch, Esperanto, Französisch, Holländisch, Italienisch, Lettisch, Litauisch, Russisch und Spanisch. Andere Sprachen können problemlos ergänzt werden. Das Format der language.txt ist sehr einfach: In einzelnen Blöcken, die durch Leerzeilen getrennt sind, wird je Zeile eine Übersetzung angegeben. Hierzu steht dort zunächst die US-englische Meldung, die nicht verändert werden darf, da sie fest im Programm eingebaut ist und zum Suchen der Übersetzung verwendet wird, anschließend folgen in einzelnen Zeilen die Übersetzungen. Jede Zeile beginnt mit dem Kürzel für die Sprache, das in LANGUAGE definiert ist. Ein Beispiel:

```
"\nClear signature file: %s\n"
de: "\nDateiname der Klartext-Unterschrift: %s\n"
es: "\nFichero normal con firma: %s\n"
fr: "\nFichier de signature en clair: %s\n"

"Invalid ASCII armor header line: \"%.40s\"\n\
ASCII armor corrupted.\n"
de: "\nUnzulässige Kopfzeile \"%.40s\"\n\
in der ASCII-Versandhülle. \
Die Versandhülle ist deshalb ungültig.\n"
es: "Línea incorrecta en la cabecera \
de la armadura ASCII:\n\
\"%.40s\"\n\
```

```
Armadura dañada\n"
fr: "Entête enveloppe ASCII invalide: \"%.40s\"\n\
l'enveloppe ASCII est corrompue"
```

Mehrzeilige Textmeldungen können auch verwendet werden, wie das Beispiel zeigt. Stellen, an denen PGP Texte oder Zahlen einsetzt, werden – wie in C üblich – durch `%s` oder ähnliche Konstrukte markiert. Es ist zu empfehlen, diese bei einer Übersetzung direkt zu übernehmen; falls mehrere vorhanden sind, darf die Reihenfolge nicht geändert werden.

Mit dem Parameter `LANGUAGE` wird festgelegt, in welcher Sprache die Meldungen angezeigt werden sollen. Für `LANGUAGE` sind folgende Werte definiert:

`en`	für US-Englisch	`es`	für Spanisch
`de`	für Deutsch	`nl`	für Holländisch
`fr`	für Französisch	`it`	für Italienisch
`ru`	für Russisch	`lt3`	für Litauisch
`lv`	für Lettisch	`esp`	für Esperanto.

Bei der Einstellung

```
LANGUAGE = fr
```

würden beispielsweise die Texte auf Französisch erscheinen – vorausgesetzt, `language.txt` enthält französische Texte.

Wenn PGP eine Meldung am Bildschirm anzeigen muß, sucht es in der Datei `language.txt` nach dem Text in der gewählten Sprache. Falls PGP die Datei nicht findet, oder wenn Texte in der gewählten Sprache in `language.txt` nicht vorhanden sind, oder wenn eine einzelne Meldung nicht übersetzt ist, wird der englische Text ausgegeben.

Um die Distribution klein zu halten, sind die meisten Übersetzungen nicht im PGP-Paket vorhanden, sondern getrennt erhältlich.

Sollen auch die Hilfetexte von PGP (Kommando `pgp -h`) in einer anderen Sprache als US-Englisch ausgegeben werden, muß außerdem eine Datei mit dem Suffix `.hlp` existieren, wobei der Name der Datei für einen der oben genannten Werte steht, also z. B. `de.hlp` für Deutsch.

GnuPG ist auf die modernere Art „lokalisiert“, es wertet die Umgebungsvariable `LANG` aus und verwendet Textmeldungen, die im Standardpfad für lokalisierte Textmeldungen untergebracht sind. Auf meinem System ist das `/usr/share/locale/<Sprache>/LC_MESSAGES/`. Bei der Installation werden hier Sprachdateien für Deutsch (`de`), Brasiliani-

sches Portugiesisch (pt_BR), Spanisch (es), Französisch (fr), Italienisch (it), Polnisch (pl) und Russisch (ru) installiert, US-Englisch ist im Programm eingebaut.

MYNAME – Standard-Benutzer-ID für Unterschriften

Standardeinstellung: MYNAME = ""
GnuPG: default-key

Mit MYNAME/default-key kann ausgewählt werden, welchen geheimen Schlüssel PGP automatisch für Unterschriften wählt. Wenn MYNAME nicht definiert ist, wird der neueste geheime Schlüssel verwendet. Wenn die Option -u Benutzer-ID beim Aufruf von PGP angegeben wird, hat diese Auswahl Vorrang vor der Auswahl durch MYNAME.

TEXTMODE – Standardmäßig Text verschlüsseln

Standardeinstellung: TEXTMODE = off
GnuPG: textmode

Der Parameter TEXTMODE ist äquivalent zu der Kommandozeilen-Option -t. Wenn TEXTMODE=on gewählt wird,[∨] geht PGP davon aus, daß die zu verschlüsselnden Daten keine Binärdaten, sondern Text sind. In diesem Fall werden die Daten vor der Verschlüsselung in eine kanonische, also systemunabhängige, Form konvertiert. Text in kanonischer Form verwendet als Zeichensatz latin1 (bzw. im neuen Datenformat utf-8) und als Zeilentrennung die Zeichen Wagenrücklauf und Zeilenvorschub. Näheres zur Konvertierung finden Sie im folgenden Abschnitt. PGP schaltet die Konvertierung in kanonische Form automatisch aus, wenn es Daten erkennt, die es für Binärdaten hält.

CHARSET – Der Zeichensatz Ihres Computers

Standardeinstellung: CHARSET = NOCONV
GnuPG: charset

PGP kann die Sonderzeichen vieler Sprachen für die Zeichensätze verschiedener Computer konvertieren. Damit dies korrekt funktioniert,

∨ Für GnuPG: Wenn in der Konfigurationsdatei eine Zeile textmode steht ...

müssen Sie den Parameter CHARSET richtig einstellen. Diese Konvertierung erfolgt nur bei der Verschlüsselung von Textdateien. Falls Sie mit PGP ausschließlich Texte ver- und entschlüsseln, die nur Buchstaben des „normalen Alphabets“, also keine Umlaute, Buchstaben mit Akzenten oder anderen diakritischen Zeichen enthalten, ist der Parameter CHARSET für Sie unwichtig. Das dürfte aber kaum der Fall sein, wenn Sie deutschsprachige Texte schreiben oder empfangen – der Zeichensatz reicht nicht einmal für englische Texte aus, beispielsweise das Wort „*naïve*“ läßt sich damit nicht schreiben. Daher sollten Sie CHARSET korrekt setzen, damit auch Empfänger mit einem anderen Betriebssystem Ihre Nachrichten korrekt lesen können. CHARSET teilt PGP mit, welchen Zeichensatz Ihr Computer verwendet.

Wert	Bedeutung
NOCONV	keine Konvertierung
LATIN1	ISO 8859-1 Lateinisches Alphabet 1
KOI8	verwendet auf vielen russischen Unix-Anlagen
ALT_CODES	verwendet auf russischen MS-DOS-Computern
ASCII	7-Bit-Zeichensatz ohne Umlaute
CP850	MS-DOS, für Deutsch, Französisch etc.

Tabelle 14.1: In PGP 2.6.x definierte Werte für CHARSET

CHARSET kann bei PGP 2.6.x die in Tabelle 14.1 angegebenen Werte annehmen. PGP 2.6.x verwendet für die interne kanonische Textdarstellung latin1. Wenn also CHARSET=LATIN1 gewählt wird, findet keine Zeichenkonvertierung statt. Zu beachten ist, daß PGP auch KOI8 wie LATIN1 behandelt, obwohl KOI8 für einen völlig anderen Zeichensatz (kyrillisch) steht. Eine Konvertierung von KOI8 in LATIN1 oder CP850 wäre aber sinnlos. Die Einstellungen NOCONV, LATIN1 und KOI8 sind für PGP äquivalent.

Wenn Sie mit MS-DOS arbeiten und Nachrichten verschicken oder erhalten, die in einer westeuropäischen Sprache geschrieben sind, sollten Sie CHARSET=CP850 einstellen. Wenn Sie dann eine Nachricht mit der Option -t oder TEXTMODE=on verschlüsseln, konvertiert PGP Ihren Text vor der Verschlüsselung in den LATIN1-Zeichensatz. Bei der Entschlüsselung wird LATIN1 in CP850 umgewandelt.

Für Windows stellen Sie bitte CHARSET=latin1 ein.

Für GnuPG gilt fast dasselbe, hier sind aber bislang nur die Zeichensätze iso-8859-1 (westeuropäische Sprachen, default, entspricht latin1), iso-8859-2 (osteuropäische Sprachen) und koi8-r definiert, außerdem werden im neuen Datenformat alle Texte in das Unicode-Format utf-8 gebracht, womit ein verlustfreier Transport einer großen Menge von Sprachen (einschließlich aller europäischen Sprachen und eines Großteils der japanischen Standard-Schriftzeichen) möglich ist, ohne daß reiner ASCII-Text irgendwie aufgebläht werden müßte.

ARMOR – ASCII-Darstellung verschlüsselter Dateien

Standardeinstellung: ARMOR = off
GnuPG: armor

Der Parameter ARMOR ist äquivalent zur Kommandozeilenoption -a. Wenn ARMOR=on gewählt wird,⊕ stellt PGP die verschlüsselten Daten im Radix-64-Format dar. Dieses Format ist für den Versand über manche E-Mail-Kanäle sinnvoll. Die von PGP erzeugten Dateien im Radix-64-Format haben das Suffix .asc.

Es ist vermutlich sinnvoll, ARMOR=on zu wählen. Weiteres hierzu steht im Abschnitt 13.10.

ARMORLINES – maximale Größe von ASCII-dargestellten Dateien

Standardeinstellung: ARMORLINES = 720

Wenn PGP eine sehr große Datei im Radix-64-Format erzeugen soll, teilt es diese Datei in mehrere Dateien auf, die jeweils klein genug sind, um im Internet versandt zu werden.

Der Parameter ARMORLINES gibt an, wieviele Zeilen eine von PGP erzeugte .asc-Datei maximal enthalten darf. Wird ARMORLINES auf 0 gesetzt, kann eine .asc-Datei beliebig groß werden.

Viele Mail-Transport-Programme im Internet lassen keine Nachrichten zu, die mehr als etwa 50000 Byte groß sind. Eine Datei mit 720 Zeilen im Radix-64-Format liegt weit genug unter dieser Grenze, um problemlos versandt werden zu können. Die einzelnen Dateien, die PGP erzeugt, erhalten als Suffix .as1, .as2, .as3 usw.

⊕ bzw. eine Zeile armor vorhanden ist

GnuPG erzeugt immer nur eine einzelne Ausgabedatei, da heutzutage die meisten E-Mail-Programme dazu in der Lage sind, überlange Nachrichten mit Hilfe des MIME-Standards zu zerteilen.

KEEPBINARY – verschlüsselte Binärdatei nach Entschlüsselung nicht löschen

Standardeinstellung: `KEEPBINARY = off`

Wenn PGP eine `.asc`-Datei einliest, erkennt es automatisch, daß es eine Datei im Radix-64-Format ist, und konvertiert sie zurück in ihre binäre Form (also eine `.pgp` Datei), bevor es mit der eigentlichen Entschlüsselung beginnt. Bei der Entschlüsselung erzeugt PGP natürlich auch eine Datei mit dem Klartext.

PGP ermöglicht die Auswahl, ob man die `.pgp`-Datei behalten möchte, oder ob sie nach der Entschlüsselung gelöscht werden soll. Die `.asc`-Datei bleibt in jedem Fall erhalten.

Wenn `KEEPBINARY=on` eingestellt wird, bleibt die `.pgp`-Datei erhalten; wird `KEEPBINARY=off` eingestellt, wird die `.pgp`-Datei nach der Entschlüsselung gelöscht.

Mit GnuPG können Sie die ASCII-Transportverpackung so entfernen:

```
gpg: gpgm --dearmor datei.asc
```

COMPRESS – Datenkompression ein- oder ausschalten

Standardeinstellung: `COMPRESS = on`

Mit `COMPRESS=on/off` kann eingestellt werden, ob PGP den Klartext vor der Verschlüsselung komprimiert. `COMPRESS=off` ist im wesentlichen für das Debuggen von PGP interessant; in der Regel sollte `COMPRESS=on` gewählt werden, damit PGP den Klartext vor der Verschlüsselung komprimiert.

Sollten Sie diese Option bei GnuPG wirklich brauchen, können Sie auf der Kommandozeile `-z 0` hinzufügen.

COMPLETES_NEEDED – Anzahl der erforderlichen voll vertrauenswürdigen Unterschriften

Standardeinstellung: `COMPLETES_NEEDED = 1`

GnuPG: completes-needed

Mit COMPLETES_NEEDED läßt sich einstellen, wieviele voll vertrauenswürdige Unterschriften PGP unter einem Schlüssel verlangt, um diesen Schlüssel als vollständig bestätigt zu betrachten. Mit diesem Parameter hat man also die Möglichkeit, PGP mehr oder weniger mißtrauisch einzustellen. Genaueres hierüber finden Sie im Abschnitt 7.3.

MARGINALS_NEEDED – Anzahl der erforderlichen teilweise vertrauenswürdigen Unterschriften

Standardeinstellung: MARGINALS_NEEDED = 2
GnuPG: marginals-needed

Mit MARGINALS_NEEDED läßt sich einstellen, wieviele teilweise vertrauenswürdige Unterschriften PGP unter einem Schlüssel verlangt, damit dieser Schlüssel als vollständig bestätigt betrachtet wird. Mit diesem Parameter hat man also die Möglichkeit, PGP mehr oder weniger mißtrauisch einzustellen. Genaueres hierüber finden Sie im Abschnitt 7.3. Der Standardwert bei GnuPG ist 3.

CERT_DEPTH – Schachtelungstiefe von Unterschriften

Standardeinstellung: CERT_DEPTH = 4
GnuPG: max-cert-depth

CERT_DEPTH gibt an, bis zu welcher Tiefe PGP Unterschriften unter öffentliche Schlüssel prüft, d. h. wie „indirekt“ die Bestätigung der Echtheit eines Schlüssels sein darf. Wenn Sie beispielsweise CERT_DEPTH=1 wählen, erkennt PGP nur solche Schlüssel als voll bestätigt an, die von einer Person unterschrieben sind, deren öffentlichen Schlüssel Sie persönlich mit Ihrem geheimen Schlüssel unterschrieben haben. Setzen Sie CERT_DEPTH=0, erkennt PGP nur die Unterschriften als voll vertrauenswürdig, die Sie selbst geleistet haben. Wenn Sie CERT_DEPTH=2 setzen, ist für PGP auch der Schlüssel von Carol voll bestätigt, wenn Carols Schlüssel von Bob, Bobs Schlüssel von Alice, und Alices Schlüssel von Ihnen selbst unterschrieben ist und Sie Alices und Bobs Schlüssel als vertrauenswürdig erklärt haben.

Der Standardwert für GnuPG beträgt 5, der kleinste zulässige Wert für `CERT_DEPTH` ist `0`, der größte `8`. Genaueres hierüber finden Sie im Abschnitt 7.3.

`BAKRING` – Dateiname der Sicherheitskopie des Bundes mit geheimen Schlüsseln

Standardeinstellung: `BAKRING = ""`

Die Prüfung der Echtheit eines öffentlichen Schlüssels durch Unterschriften basiert letztlich auf der Echtheit Ihres eigenen Schlüssels, den PGP als absolut vertrauenswürdig ansieht. (Sie können auch mehrere eigene Schlüssel haben, die PGP als voll vertrauenswürdig anerkennt.)

Um mögliche Fälschungen an Ihrem Bund mit öffentlichen Schlüsseln erkennen zu können, muß PGP kontrollieren, ob auch Ihr eigener Schlüssel nicht gefälscht wurde. Hierfür vergleicht PGP Ihren öffentlichen Schlüssel mit einer Sicherheitskopie Ihres geheimen Schlüssels, die auf einem fälschungssicheren Medium, beispielsweise einer schreibgeschützten Diskette, gespeichert ist. Im geheimen Schlüssel sind alle Informationen gespeichert, die Ihr öffentlicher Schlüssel hat (aber nicht die Unterschriften darunter). Dies bedeutet, daß PGP Ihren öffentlichen Schlüssel mit der Sicherheitskopie Ihres geheimen Schlüssels vergleichen kann.

Mit dem Parameter `BAKRING` können Sie den Pfadnamen festlegen, unter dem PGP die Sicherheitskopie Ihres geheimen Schlüssels sucht. (Beispielsweise können Sie unter MS-DOS/Windows mit einer Einstellung der Art `BAKRING=a:\secring.pgp` erreichen, daß PGP die Sicherheitskopie auf einer (hoffentlich schreibgeschützten) Diskette sucht. Diese müßten Sie dann bei Bedarf immer einlegen.) Diesen Vergleich mit der Sicherheitskopie führt PGP nur durch, wenn Sie mit `pgp -kc` Ihren gesamten Bund mit öffentlichen Schlüsseln prüfen (vgl. Abschnitt 15.14). GnuPG bietet leider keine derartige Prüfung.

Wenn `BAKRING` nicht definiert ist, führt PGP diese Kontrolle Ihres eigenen öffentlichen Schlüssel nicht durch.

`PUBRING` – Dateiname des Bundes mit öffentlichen Schlüsseln

Standardeinstellung: `PUBRING = $PGPPATH/pubring.pgp`
GnuPG: `keyring`

Dies legt den Namen der Datei fest, in der PGP die öffentlichen Schlüssel sucht. $PGPPATH wird von PGP durch die Umgebungsvariable PGPPATH ersetzt. Bei GnuPG lassen sich mit dieser Angabe beliebig viele Schlüsselbunde angeben, der Default-Schlüsselbund läßt sich z. B. so festlegen:

```
keyring gnupg-ring:~/.gnupg/pubring.gpg
keyring bigring.gpg
```

SECRING – Dateiname des Bundes mit privaten Schlüsseln

Standardeinstellung: `SECRING = $PGPPATH/secring.pgp`
GnuPG: `secret-keyring`

Analog zu PUBRING legt SECRING den Namen der Datei mit privaten (geheimen) Schlüsseln fest. Für GnuPG gelten die Kommentare aus dem vorangegangenen Abschnitt analog.

RANDSEED – Dateiname der Datei für die Zufallszahlen

Standardeinstellung: `RANDSEED = $PGPPATH/randseed.bin`

Diese Einstellung bezeichnet die Datei, die PGP als „Pool“ für die Zufallszahlen dient. Diese Datei wird von PGP nach Generierung der Zufallszahlen verschlüsselt, um einem möglichen Angriff vorzubeugen. Wenn Sie den Zufallszahlen von PGP nicht vertrauen, können Sie die Datei nach jeder Benutzung löschen – dann müssen Sie aber immer mit Tastatureingaben für neue Zufallszahlen sorgen. Näheres finden Sie in Abschnitt 4.3. GnuPG verwendet immer vom Betriebssystem bereitgestellte Zufallszahlen, so daß für diesen Parameter keine Verwendung mehr besteht.

Achtung: Wenn PGP 5.0 nicht mit einer Quelle wie /dev/random arbeitet, muß die Datei randseed.bin unbedingt an ihrem im Programm eingebrannten Platz (also $PGPPATH/randseed.bin) bleiben – ansonsten *verschlüsselt PGP jede Nachricht mit demselben Wegwerf-Schlüssel,* was im Endeffekt fast dasselbe ist wie gar keine Verschlüsselung. Das ist natürlich kein beabsichtigtes Feature, sondern ein schwerwiegender Programmfehler.

PAGER – Auswahl eines Programms für die Textanzeige am Bildschirm

Standardeinstellung: `PAGER = ""`

Wenn Sie beim Entschlüsseln die Option `-m` angeben, können Sie den entschlüsselten Text am Bildschirm lesen, ohne daß PGP ihn in eine Datei schreibt. Standardmäßig verwendet PGP hierzu eigene Routinen, die ähnlich dem Programm `more` bei Unix arbeiten.

Falls Sie ein anderes Programm für Textanzeige am Bildschirm bevorzugen, können Sie es unter `PAGER` eintragen. Unter Unix bietet sich `less` an, unter MS-DOS können Sie beispielsweise das populäre Programm `LIST` von VERNON D. BUERG verwenden. In diesem Fall würde der `PAGER`-Eintrag so lauten:

```
PAGER = list
```

Wenn jedoch die Absenderin einer Nachricht die Option `-m` (Klartext nach Entschlüsseln nicht in eine Datei schreiben) angegeben hat, verwendet PGP in jedem Fall seine eigene Anzeigefunktion, da die meisten „pager“ auch eine Funktion zum Editieren oder Speichern eines Textes haben. GnuPG verwendet selbst keinen „pager“, so daß die Option keinen Sinn macht.

SHOWPASS – Anzeige des Mantra während der Eingabe

Standardeinstellung: `SHOWPASS = off`

Normalerweise zeigt PGP das Mantra während der Eingabe nicht am Bildschirm an. Dadurch wird es für neugierige Augen schwerer, das Mantra mitzulesen. Es gibt aber Menschen, die Schwierigkeiten haben, ihr Mantra korrekt einzutippen, ohne daß sie es am Bildschirm sehen können. So entstand der Wunsch, PGP für die Anzeige des Mantra konfigurierbar zu machen. In der Abgeschiedenheit einer Wohnung ist es nicht allzu problematisch, das Mantra am Bildschirm anzeigen zu lassen. Wird `SHOWPASS=on` eingestellt, zeigt PGP das Mantra beim Eintippen am Bildschirm an.

TZFIX – Zeitzonenkorrektur

Standardeinstellung: `TZFIX = 0`

PGP versieht Schlüssel und Unterschriften mit einer Zeitmarke. Hierzu wird intern Coordinated Universal Time (UTC) (oder Greenwich Mean Time (GMT), was für unsere Zwecke dasselbe ist,) verwendet. GMT ist die Zeit in Großbritannien, ohne Sommerzeit.

Die Kurzfassung des Folgenden für Ungeduldige: Wenn beim Aufruf

```
pgp
```

eine falsche GMT-Uhrzeit angezeigt wird, versuchen Sie es unter MS-DOS zunächst mit dem Befehl

```
SET TZ=MET-1DST
pgp
```

und wenn das noch nicht hilft, setzen Sie `TZFIX` auf die Abweichung, die die angezeigte Zeit von der echten Zeit hat. PGP kommt leider nur mit Zeitzonen mit ganzzahligen Abweichungen von GMT klar. So sollte das Ergebnis beispielsweise während der Sommerzeit aussehen:

```
[ccr@nescio pgp_doku]$ date
Tue Sep 28 11:55:52 CEST 1999
[ccr@nescio pgp_doku]$ pgp
...
Aktuelles Datum und Uhrzeit: 1999/09/28 09:56 GMT
...
```

Ab jetzt folgt die Langversion, die Sie nur bei Einstellungsproblemen brauchen sollten und auch dann nur ein einziges mal:

Wenn PGP das Betriebssystem nach Datum und Zeit fragt, nimmt es an, daß die Zeit als GMT-Zeit zurückgegeben wird. Unter Umständen wird die Zeit aber auf schlecht konfigurierten MS-DOS-Rechnern als US Pacific Standard Time interpretiert und daher die lokale Zeit plus acht Stunden zurückgegeben. Seltsam, nicht wahr? Vielleicht liegt es an einer Art US-Westküsten-Chauvinismus, daß MS-DOS bzw. der Borland Compiler davon ausgeht, die lokale Zeit sei die US Pacific Time, und GMT darauf basierend ausrechnet. Dies wirkt sich nachteilig auf das Verhalten der MS-DOS-internen Funktionen aus, die PGP verwendet.

Wenn jedoch die MS-DOS Umgebungsvariable `TZ` für Ihre Zeitzone korrekt definiert ist, korrigiert dies auch die irrtümliche Annahme von MS-DOS, die ganze Welt lebe an der Westküste der USA.

`TZFIX` gibt die Anzahl der Stunden an, die PGP zur „Betriebssystemzeit“ addiert, um GMT-Zeitangaben für Unterschriften und Schlüssel zu erhalten. Wenn die Betriebssystemzeit korrekt ist, also bspw. die MS-DOS Umgebungsvariable `TZ` korrekt definiert ist, kann `TZFIX` auf dem Standardwert `0` bleiben. Unter Unix ist `TZFIX` in der Regel auch nicht notwendig. `TZFIX` kann aber für irgendwelche obskuren Betriebssysteme sinnvoll sein, die nie etwas von GMT gehört haben.

In Los Angeles:	`SET TZ=PST8PDT`	In London:	`SET TZ=GMT0BST`
In Denver:	`SET TZ=MST7MDT`	In Bielefeld:	`SET TZ=MET-1DST`
In Arizona:	`SET TZ=MST7`	In Moskau:	`SET TZ=MSK-3MSD`
In Chicago:	`SET TZ=CST6CDT`	In Auckland:	`SET TZ=NZT-13`
In New York:	`SET TZ=EST5EDT`		

Eine wesentlich sauberere Lösung ist die Definition der MS-DOS-Umgebungsvariablen `TZ` in der `autoexec.bat`. In diesem Fall liefert MS-DOS die korrekte GMT-Zeit und berücksichtigt auch die Sommerzeit, abhängig von der jeweiligen Zeitzone. (Die Einstellung für Bielefeld gilt auch für den Rest Deutschlands, für die Schweiz, Österreich, die Niederlande, Frankreich etc.)

Die ersten drei Zeichen des Wertes von `TZ` müssen Buchstaben sein, danach muß eine ein- oder zweistellige Zahl, ggf. mit einem Minuszeichen davor, stehen. Stehen hinter der Zahl noch Buchstaben, wertet `gmtime()` dies als Signal, daß es eine Sommerzeit gibt. Welche Buchstaben vor und ggf. hinter der Zahl stehen, wertet `gmtime()` nicht weiter aus.

Bei der Sommerzeit-Option ist zu beachten, daß `gmtime()` unter MS-DOS (und evtl. noch weiteren Systemen) als Beginn und Ende der Sommerzeit die in den USA zutreffenden Tage verwendet, die glücklicherweise inzwischen auch in Europa gelten. Wie es in anderen Zeitzonen aussieht, entzieht sich meiner Kenntnis.

CLEARSIG – Nachrichten im Klartext mit ASCII-Unterschrift

Standardeinstellung: CLEARSIG = on

Eine Erklärung hierzu finden Sie weiter vorne in Abschnitt 13.11. Die Option in config.txt wirkt sich nur dann aus, wenn eine Textnachricht nur signiert (also nicht verschlüsselt) wird und das Ergebnis auch (per -a oder ARMOR-Eintrag in config.txt) ASCII-kodiert ausgegeben werden soll. Das kann paradoxerweise dazu führen, daß eine Nachricht, die laut Kommandozeile mit ASCII-Verpackung versandt werden sollte, doch Umlaute enthält, die evtl. auf dem Transportweg beschädigt werden.

VERBOSE – keine, normale oder ausführliche Meldungen

Standardeinstellung: VERBOSE = 1
GnuPG: verbose

VERBOSE kann auf 0, 1 oder 2 gesetzt werden, je nachdem, wie detailliert die Meldungen von PGP sein sollen:

0 Meldungen werden nur ausgegeben, wenn es Probleme gibt. Das richtige für Unix-Fans.

1 Die Standardeinstellung. PGP zeigt in sinnvollem Umfang diagnostische Meldungen und Bedienungshinweise.

2 Ausführliche Meldungen. Diese Option ist hauptsächlich für die Fehlersuche gedacht. Normalerweise ist sie nicht sinnvoll.

Die Option verbose für GnuPG erhält kein Argument, sie kann aber zweimal gesetzt werden, um das Äquivalent zu VERBOSE=2 zu erhalten.

INTERACTIVE – Bestätigungsabfrage beim Hinzufügen von öffentlichen Schlüsseln

Standardeinstellung: INTERACTIVE = off

Wenn INTERACTIVE=on gesetzt wird, fragt PGP beim Bearbeiten einer Datei, die mehrere öffentliche Schlüssel enthält, für jeden Schlüssel einzeln nach, ob er aufgenommen werden soll.

NOMANUAL – Erzeugung von Schlüsseln zulassen, ohne daß ein Handbuch auf der Festplatte vorhanden ist

Standardeinstellung: `NOMANUAL = off`

Es ist wichtig, daß PGP nur zusammen mit den Handbuchdateien, die zum normalen PGP-Distributionspaket gehören, vertrieben wird. Das Handbuch enthält wichtige Informationen zur Bedienung von PGP, sowie wichtige rechtliche Hinweise. Manche Leute haben aber ältere Versionen von PGP ohne das Handbuch vertrieben, was bei den Leuten, die dieses „Vertriebspaket“ bekamen, zu einer Reihe von Problemen führte. (Bitte beachten Sie hierzu auch den Abschnitt E.3 auf Seite 285.) Um die Weitergabe von PGP ohne Dokumentation zu unterbinden, wurde PGP so modifiziert, daß es prüft, ob die Handbuchdateien irgendwo auf dem Computer vorhanden sind (z. B. im PGP-Verzeichnis), bevor es die Erzeugung eines Schlüsselpaares zuläßt.

Manche Menschen verwenden PGP aber auf winzigen Palmtop-Rechnern mit sehr beschränkter Speicherkapazität. Hier kann es sinnvoll sein, die Handbuchdateien von der Festplatte zu löschen. Um diesen Anwenderinnen gerecht zu werden, kann PGP mit Hilfe der `NOMANUAL`-Option so eingestellt werden, daß es Schlüsselgenerierung auch dann zuläßt, wenn es die Handbuchdateien nicht findet. Dies geschieht mit der `NOMANUAL`-Option beim Kommando für die Schlüsselerzeugung:

2.6.x: `pgp -kg +nomanual`

Die `NOMANUAL`-Option kann *nur* in der Kommandozeile angegeben werden. Folglich müssen Sie schon das Handbuch lesen, um herauszufinden, wie diese Option funktioniert. Damit steht diese Beschreibung genaugenommen im falschen Abschnitt, aber die Option wird auf der Kommandozeile genauso verwendet wie alle anderen Optionen dieses Kapitels, was ein starkes Argument dafür ist, sie hier einzusortieren. GnuPG und PGP 5.0 haben keine derartigen Einschränkungen, also auch diesen Parameter nicht.

COMMENT – Kommentar für Radix64-codierte Daten

Standardeinstellung: `COMMENT = ""`
GnuPG: `comment`

Mit COMMENT läßt sich ein beliebiger Text festlegen, der bei Verwendung von Radix-64 als „Comment:“ oder eine entsprechende Übersetzung in die zweite Zeile geschrieben wird (im Klartext).∞

EncryptToSelf – Eigenen Schlüssel implizit als Empfänger eintragen

Standardeinstellung: EncryptToSelf = off
GnuPG: encrypt-to

Mit EncryptToSelf läßt sich PGP so einstellen, daß alle ausgehenden Nachrichten zusätzlich zu den angegebenen Empfängern auch für den eigenen Schlüssel (näheres siehe Abschnitt MYNAME) verschlüsselt werden. GnuPG bietet die Konfigurationsoption encrypt-to, mit der beliebige Schlüssel angegeben werden können, an die jede Nachricht mitverschlüsselt wird.

Bitte beachten Sie, daß diese Option *nicht* verwendet werden sollte, wenn Sie anonyme Nachrichten versenden möchten, da Sie sonst durch den zusätzlichen in der Nachricht eingetragenen Empfänger Ihre Identität preisgeben. GnuPG bietet hierfür die Option --no-encrypt-to (ohne Parameter aufzurufen), für PGP können Sie +encrypttoself=off auf die Kommandozeile schreiben.

LEGAL_KLUDGE – (nicht) von 2.3a lesbare Dateien erzeugen

Standardeinstellung: LEGAL_KLUDGE = on

Mit dieser Option (der Version 2.6.2i) läßt sich PGP dazu bewegen, Dateien zu erzeugen, die von PGP-Versionen <2.6 gelesen werden können. Näheres steht in Anhang A auf Seite 264.

∞ Die ähnliche Option ARMOR_VERSION existiert nur in der Version 2.6ui, allerdings läßt sich in der Datei language.txt eine beliebige „Versionsnummer“ einstellen, die in der Transportverpackung angegeben wird.

15. Spezielle Befehle

Dieser Teil des Handbuchs beinhaltet spezielle Themen, die nicht im ersten Teil „Grundlagen“ enthalten sind. Das Lesen dieses Teils ohne Kenntnis des vorangegangenen ist nicht sehr sinnvoll. Andererseits ist die Kenntnis dieses Teils für die Benutzung von PGP nicht unbedingt erforderlich.

15.1. Auswahl eines Schlüssels über seine Schlüssel-ID

Bei allen Kommandos, die die Angabe einer Benutzer-ID oder eines Teils derselben erfordern, kann statt dessen auch die hexadezimale Schlüssel-ID benutzt werden. Für PGP muß vor der Schlüssel-ID ein `0x` geschrieben werden. Beispiel:

```
2.6.x: pgp -kv 0x67F7
```

listet alle Schlüssel auf, in denen `67F7` ein Teil der Schlüssel-ID ist.

GnuPG und PGP 5.0 erwarten die gesamte kurze ID, die gesamte lange ID oder den gesamten Fingerabdruck des Schlüssels. Beginnt der Parameter mit einem Buchstaben, muß eine `0` vorangestellt werden, PGP 5.0 erwartet auf jeden Fall ein `0x`:

```
gpg: gpg -kv 0621CC013
gpg: gpg -e datei \
        -r 8AFAB30A453B6BFD6DDAC3DA2F654DBE32106275
 5.0: pgpk -l 0xD0938CF5
```

Diese Option ist vor allem dann praktisch, wenn es von einer Person zwei verschiedene Schlüssel mit ein und derselben Benutzer-ID gibt. In diesem Fall läßt sich mit Hilfe der Schlüssel-ID einer der beiden Schlüssel eindeutig auswählen. Die Auswahl eines Schlüssels über seinen Fingerprint ist wohl ausschließlich für den automatisierten Ablauf interessant.

15.2. Trennung der Unterschrift von der Nachricht

Normalerweise wird eine Unterschrift in derselben Datei gespeichert wie der Text, der unterschrieben wird. Dadurch ist die Prüfung einer Unterschrift in den meistens Fällen einfach und bequem möglich. Unter bestimmten Umständen ist es aber sinnvoll, die Unterschrift in einer eigenen Datei, unabhängig von der Textdatei, zu speichern. Hierzu dient die Option `-b` zusammen mit der Option `-s`.

Beispiel:

```
2.6.x: pgp -sb brief.txt
  5.0: pgps -b brief.txt
  gpg: gpg -sb brief.txt
  gpg: gpg -b brief.txt
  gpg: gpg --detach-sign brief.txt
```

Hier wird eine Datei `brief.sig` bzw. `brief.txt.sig` erzeugt, die nur die Unterschrift enthält. Der Inhalt der Datei `brief.txt` wird *nicht* in `brief.sig` gespeichert (und kann aus dieser Datei auch nicht berechnet werden). Wenn die Unterschrift in einer eigenen Datei (`brief.sig` in obigem Beispiel) erzeugt wurde, müssen beide Dateien (im Beispiel `brief.sig` und `brief.txt`) an die Empfängerin geschickt werden. Sie benötigt beide Dateien, um die Echtheit der Unterschrift zu prüfen. Wenn sie die Datei mit der Unterschrift durch PGP bearbeiten läßt, stellt das Programm fest, daß diese Datei keinen Text enthält, und fragt die Benutzerin nach dem Namen der Textdatei. Erst danach kann PGP die Unterschrift prüfen. Wenn der Empfänger bereits weiß, daß Text und Unterschrift in getrennten Dateien gespeichert sind, kann er auch beide Dateinamen in der Kommandozeile angeben:

```
2.6.x: pgp brief.sig brief.txt
  5.0: pgpv brief.sig brief.txt
  gpg: gpg --verify brief.txt.sig brief.txt
```

In diesem Fall fragt PGP nicht nach dem Namen der Textdatei. Die Unterschrift in einer eigenen Datei zu speichern, ist dann sinnvoll, wenn die Unterschriften unabhängig vom Text protokolliert werden sollen, beispielsweise

- als öffentliche Unterschrift unter einem Text, der erst später veröffentlicht werden soll (Prophezeiungen von Lottozahlen o. ä.).

- für Dateien mit ausführbaren Programmen (bei MS-DOS: `.EXE` und `.COM`) oder Archive (beispielswiese `.zip`), um eine Virusprüfung durchzuführen.
- wenn mehrere Personen ein Dokument (zum Beispiel einen Vertrag) unterschreiben sollen, ohne daß die Unterschriften „verschachtelt“ werden. Die Unterschrift jeder einzelnen Person wird unabhängig von den anderen gespeichert.

PGP 2.6.x kann bei einer Datei, in der Unterschrift und Text in einer Datei stehen, die Unterschrift auch nachträglich vom Text trennen. Dies geschieht mit der Option `-b` bei der Entschlüsselung:

2.6.x: `pgp -b brief`

Hier wird `brief.pgp` entschlüsselt. Falls eine Unterschrift vorhanden ist, wird sie geprüft und in einer eigenen Datei `brief.sig` gespeichert.

Diese Option fehlt leider bei PGP 5.0 und GnuPG.

15.3. Entschlüsselung einer Nachricht und Speicherung zusammen mit einer Unterschrift

Normalerweise wollen Sie eine verschlüsselte Nachricht vollständig entschlüsseln, die Unterschrift (oder mehrere verschachtelte Unterschriften) prüfen und hierbei eine „Schicht“ nach der anderen abtrennen, bis der originale Klartext übrig bleibt. Manchmal wollen Sie aber die Nachricht nur entschlüsseln und die Unterschrift bei der Nachricht belassen.

Dies ist beispielsweise dann sinnvoll, wenn Sie die Kopie einer unterschriebenen Nachricht an eine dritte Person weiterleiten möchten, gegebenenfalls erneut verschlüsselt.

Angenommen, es kommt eine Nachricht, die Charlie unterschrieben hat, verschlüsselt mit dem öffentlichen Schüssel des Empfängers. Die Nachricht soll entschlüsselt und zusammen mit Charlies Unterschrift an Alice weitergeleitet werden, gegebenenfalls mit ihrem öffentlichen Schlüssel verschlüsselt. Mit PGP ist das kein Problem. Mit folgendem Kommando wird eine Nachricht entschlüsselt, ohne daß die Unterschriften abgetrennt werden:

2.6.x: `pgp -d brief`
gpg: `gpg -d brief`
gpg: `gpg --decrypt brief`

Hier wird die Datei `brief.pgp` entschlüsselt, und Unterschriften werden, falls vorhanden, zusammen mit dem entschlüsselten Klartext in der Ausgabedatei gespeichert. Die Ausgabedatei kann archiviert oder – gegebenenfalls wieder verschlüsselt – an eine andere Person weitergeleitet werden. PGP 5.0 scheint keine derartige Option zu bieten.

15.4. Der Austausch von Text zwischen unterschiedlichen Computern

Mit PGP kann jede Art von Klartext verschlüsselt werden, irgendwelche Binärdaten* ebenso wie Text. Wahrscheinlich wird PGP am häufigsten für die Verschlüsselung von E-Mail benutzt, so daß der Klartext aus Text-Daten besteht.

Text wird auf verschiedenen Computern leicht unterschiedlich dargestellt. Beispielsweise endet eine Zeile bei MS-DOS/Microsoft Windows mit den beiden Zeichen für Wagenrücklauf und Zeilenvorschub. Bei Unix endet eine Zeile nur mit dem Zeichen für Zeilenvorschub. Beim Macintosh endet eine Zeile mit dem Zeichen für Wagenrücklauf, ohne Zeilenvorschub. Umlaute einer MS-DOS-Textdatei werden unter Windows und Unix nicht korrekt angezeigt. Traurig, aber wahr.

Nicht verschlüsselte Textdaten werden bei E-Mail-Systemen in eine kanonische Form gebracht, wenn sie zwischen zwei Computern ausgetauscht werden. Bei kanonischer Textdarstellung besteht ein Zeilenende aus den Zeichen Wagenrücklauf und Zeilenvorschub. Beispielsweise E-Mail-Nachrichten werden beim Versand im Internet so kodiert. Außerdem werden die verwendeten Umlaute in eine allgemeinverständliche Form gebracht und das verwendete Format im „Header“ der Nachricht, wo die Steuerinformationen stehen, vermerkt. Dadurch ist es einfach, Texte zwischen verschiedenen Computern auszutauschen.

Diese automatische Anpassung der Textdarstellung an das jeweilige Betriebssystem ist bei verschlüsselten Nachrichten nicht möglich, weil der Klartext durch die Verschlüsselung dem Übertragungsprogramm verborgen bleibt. Diese Aufgabe muß das Verschlüsselungsprogramm übernehmen. Deshalb kann man bei PGP angeben, ob der Klartext als Binärdaten oder als Text angesehen werden soll. In letzterem Fall wird der Klartext vor der Verschlüsselung in eine kanonische Form gebracht.

* Beispielsweise Bilder, Programme oder Word-Dateien

Bei der Entschlüsselung wird er dann in die Form gebracht, die für das Betriebssystem des Computers, auf dem entschlüsselt wird, geeignet ist.

Wenn die Option `-t` bzw. `--textmode` beim Verschlüsseln und/oder Unterschreiben einer Nachricht mit angegeben wird, konvertiert PGP den Text vor der Verschlüsselung bzw. dem Unterschreiben in die kanonische Form.

```
2.6.x: pgp -et message.txt Empfänger-ID
  5.0: pgpe -t -r Alice message.txt
  gpg: gpg -e -r Bob --textmode message.txt
```

Diese Option schaltet PGP automatisch ab, sobald es in der zu verschlüsselnden Datei Daten findet, die es nicht als Text betrachtet. Falls der Klartext aus einem 8-Bit-Zeichensatz besteht, falls er also Zeichen enthält, die nicht im ASCII-Standard enthalten sind (Umlaute o. ä.), verwendet PGP bei der Konvertierung in die kanonische Form den Zeichensatz latin1 (ISO 8859-1 Latin Alphabet 1). Die Konvertierung hängt davon ab, was als Parameter `CHARSET` in der PGP-Konfigurationsdatei eingetragen ist, vgl. Abschnitt 14 auf Seite 106. Latin1 ist eine Obermenge von ASCII, mit diakritischen Zeichen (ä, ö, ü, ł, ð, ...) für viele westeuropäische Sprachen.

15.5. Vermeidung von „Spuren des Klartextes“ auf der Festplatte

Nachdem PGP eine Datei verschlüsselt hat, kann es bei Bedarf die Datei mit dem Klartext überschreiben und danach löschen, so daß keine „Spuren“ des Klartextes auf der Festplatte verbleiben. Dies verhindert, daß der Klartext mit einem Sektor-Editor oder einem ähnlichen Programm noch gelesen werden kann. Diese Option ist sinnvoll, um zu vermeiden, daß vertrauliche Informationen unkontrolliert auf der Festplatte verbleiben.

Um den Klartext nach der Verschlüsselung von der Festplatte zu löschen, wird die Option `w` verwendet:

```
2.6.x: pgp -esw message.txt Empfänger-ID
```

Hier wird eine verschlüsselte, unterschriebene Datei `message.pgp` erzeugt, und die Klartext-Datei `message.txt` wird danach überschrieben und gelöscht. Diese Option sollte mit Vorsicht benutzt werden – damit

gelöschte Dateien *sind* verloren, auch wenn sie nur versehentlich gelöscht wurden.

Zudem muß betont werden, daß hierdurch keinerlei Fragmente des Klartextes gelöscht werden, die ein Textverarbeitungsprogramm häufig auf der Festplatte ablegt, wenn man einen Text eintippt und bearbeitet. Die meisten Textverarbeitungsprogramme erzeugen Backup- und temporäre Dateien. Außerdem wird die Klartext-Datei nur einmal überschrieben. Das ist zwar ausreichend, um ein Lesen des Klartextes mit den üblichen Werkzeugen der Datenwiederherstellung zu verhindern, reicht aber nicht aus, um einen gezielten und ausgefeilten Leseversuch abzuwehren, bei dem eine schwache Restmagnetisierung der überschriebenen Daten mittels spezieller Hardware ausgewertet wird. Weiterhin funktioniert dieses Überschreiben nicht immer garantiert, beispielsweise bei Verwendung einer komprimierten Festplatte kann es sehr leicht fehlschlagen. PGP 5.0 und GnuPG bieten leider keine derartige Option.

15.6. Import direkt vom Keyserver

Um einen Schlüssel (bei aktiver Internet-Verbindung) direkt von einem Keyserver zu laden, können Sie die folgenden Kommandos verwenden:

```
 5.0: pgpk -a User-ID
 5.0: pgpk --HTTPKeyServerHost=horowitz.surfnet.nl \
        -a Zimmermann
 5.0: pgpk -a hkp://keys.pgp.com/user@irgend.wo
 gpg: gpg --recv-keys User-ID [User-ID2 ...]
 gpg: gpg --recv-keys --keyserver=horowitz.surfnet.nl \
        Zimmermann
```

Bei PGP 5.0 funktioniert dieser Aufruf nur dann, wenn keine Datei mit dem angegebenen Namen gefunden wird. Außerdem ist es notwendig, daß in der Konfigurationsdatei ein Keyserver korrekt eingestellt ist (oder auf der Kommandozeile angegeben wird, wie im jeweils zweiten Beispiel). Für PGP 5.0 kann dies z. B. durch einen Eintrag

```
AutoServerFetch=1
HTTPKeyServerHost=pgpkeys.mit.edu
HTTPKeyServerPort=11371
```

geschehen (was aber die Standardeinstellungen sind). GnuPG verwendet immer den Standardport 11371, hier lautet die analoge Zeile in der Datei `options`

```
keyserver pgpkeys.mit.edu
```

15.7. Export zum Keyserver

Die im vorangegangenen Abschnitt genannten Einstellungen sind für GnuPG auch hier notwendig. PGP 5.0 erfordert die explizite Angabe des Servers, daher sind keine besonderen Einstellungen vonnöten.

```
gpg: gpg --send-keys User-ID [User-ID2 ...]
 5.0: pgpk -x User-ID -o hkp://horowitz.surfnet.nl
```

15.8. Anzeige des entschlüsselten Klartextes am Bildschirm

Um den entschlüsselten Klartext nur am Bildschirm zu lesen (ähnlich dem Unix-Kommando `more`), ohne daß er in eine Datei geschrieben wird, kann die Option `-m` verwendet werden:

```
2.6.x: pgp -m brief.pgp
  5.0: pgpv -m brief.pgp
  gpg: gpg --decrypt -o- brief.pgp |less
```

Dieser Befehl zeigt den entschlüsselten Klartext am Bildschirm an. Im Abschnitt 14 auf Seite 118 erfahren Sie, wie Sie ein externes Anzeigeprogramm einbinden, das das Lesen komfortabler macht.

15.9. Verschlüsseln einer Nachricht „nur für die Augen der Empfängerin“

Um festzulegen, daß die Empfängerin den entschlüsselten Klartext nur am Bildschirm lesen, ihn aber nicht in eine Datei schreiben kann, wird die Option `-m` beim Verschlüsseln verwendet:

```
2.6.x: pgp -sem message.txt Empfänger-ID
```

Wenn die Empfängerin eine so verschlüsselte Nachricht mit ihrem privaten Schlüssel und ihrem Mantra entschlüsselt, wird der Klartext nur auf ihrem Bildschirm angezeigt, aber nicht auf der Festplatte gespeichert. Die Textanzeige erfolgt auf dem Bildschirm so, wie zuvor im Abschnitt 15.8 auf der vorherigen Seite beschrieben.

Wenn der Empfänger die Nachricht ein zweites Mal lesen will, muß er die Nachricht erneut entschlüsseln.

Diese Option ist der sicherste Weg, um zu verhindern, daß vertrauliche Nachrichten versehentlich als Klartext auf der Festplatte des Empfängers liegenbleiben, leider kann sie es aber nicht verhindern, sondern entspricht mehr einem Warnschild. GnuPG bietet leider keine derartige Option; PGP 5.0 beachtet sie zwar bei mit 2.6.x verschlüsselten Dateien, bietet aber selbst keine Möglichkeit, eine Datei in dieser Weise zu verschlüsseln.

15.10. „Anonym“ verschlüsseln

Wenn Sie Nachrichten versenden wollen, ohne daß Unbeteiligte (Mafia, Geheimdienste, Systemadministratoren, ...) erfahren, an wen Sie die Nachrichten verschlüsseln, ist einer der Schritte, Systeme wie Remailer und Mixmaster zu verwenden (siehe Kapitel 5.9 auf Seite 44 auf Seite 44). Soll nicht bekannt werden, daß der Empfänger eine verschlüsselte Nachricht erhalten hat, können Sie die entsprechenden Nachrichten beispielsweise in die Newsgroup `alt.anonymous.messages` posten. Das reicht aber nicht aus, da normalerweise in einer verschlüsselten Nachricht vermerkt wird, wer sie entschlüsseln kann. Im OpenPGP-Standard ist deshalb vorgesehen, daß an der entsprechenden Stelle auch die Schlüssel-ID `0` eingetragen werden darf, was „keine Angabe“ bedeutet. GnuPG bietet hierfür die Option `--throw-keyid`.

```
gpg: gpg -o- --throw-keyid -ear padeluun secret \
        | inews -S -ttest -nalt.anonymous.messages
```

15.11. Beibehaltung des originalen Dateinamens des Klartextes

Normalerweise gibt PGP der Datei mit dem entschlüsselten Klartext den gleichen Namen wie der Datei mit dem verschlüsselten Text, aber ohne

Suffix. Ein anderer Name für die Klartext-Datei kann mit der Option `-o` festgelegt werden.⊘ Bei den meisten E-Mail-Nachrichten ist dies sinnvoll, weil man so den Dateinamen bei der Entschlüsselung festlegen kann und typische Nachrichten meist unbrauchbare ursprüngliche Dateinamen wie `an_phil.txt` oder `krypt.msg` haben.

Wenn PGP eine Klartext-Datei verschlüsselt, fügt es den originalen Dateinamen dem Klartext vor der Komprimierung bei. Normalerweise verwendet PGP diesen originalen Dateinamen bei der Entschlüsselung nicht, aber bei Bedarf kann PGP angewiesen werden, der entschlüsselten Klartext-Datei diesen Namen zu geben. Das ist sinnvoll, wenn PGP dazu benutzt wird, Dateien zu ver- und entschlüsseln, deren Name von Bedeutung ist.

Um den originalen Namen der Klartext-Datei zu erhalten, kann die Option `-p` bzw. `--use-embedded-filename` verwendet werden. Um beim Verschlüsseln einen anderen Namen als Originalnamen einzutragen, bietet GnuPG die Option `--set-filename`.

```
2.6.x: pgp -p verschlüsselte-datei
 gpg: gpg --use-embedded-filename verschlüsselte-datei
 gpg: gpg --set-filename nochnemail -er tom /etc/passwd
```

15.12. Ändern der Benutzer-ID und des Mantras

Ab und zu kann es erforderlich sein, das Mantra zu ändern, beispielsweise dann, wenn jemand beim Eintippen des Mantras zugeschaut hat. Oder die Benutzer-ID muß geändert werden, sei es wegen einer Heirat oder wegen einer geänderten E-Mail-Adresse. Oder es soll dem Schlüssel eine zweite oder dritte Benutzer-ID hinzugefügt werden, um mehrere E-Mail-Adressen oder eine Berufsbezeichnung einzutragen.

Mit PGP können einem Schlüssel mehrere Benutzer-IDs hinzugefügt werden, und jede dieser IDs kann für die Auswahl des Schlüssel aus dem Bund mit öffentlichen Schlüsseln verwendet werden. Die eigene Benutzer-ID und das Mantra können mit folgendem Kommando geändert werden:

```
2.6.x: pgp -ke Benutzer-ID [Schlüsselbund]
  5.0: pgpk -e Benutzer-ID
```

⊘ Bei der Version 2.6.3i wird das Suffix dieses Namens evtl. überschrieben, wenn PGP meint, den Typ der entschlüsselten Datei zu kennen. Caveat emperor.

gpg: `gpg --edit-key Benutzer-ID`

PGP fragt dann nach der neuen `Benutzer-ID` und dem neuen Mantra. Wenn der optionale Parameter `Schlüsselbund` angegeben wird, muß es sich um einen Bund mit öffentlichen Schlüsseln handeln, nicht mit geheimen. Der Parameter `Benutzer-ID` muß die eigene ID sein. PGP erkennt dies daran, daß diese ID sowohl im Bund mit öffentlichen Schlüsseln als auch im Bund mit geheimen Schlüsseln auftaucht. Beide Dateien werden geändert, auch wenn ein Bund mit öffentlichen Schlüsseln als Parameter angegeben wurde.

GnuPG bietet bei diesem Aufruf ein Menü, in dem Sie u. a. die Befehle `adduid`, `deluid`, `addkey`, `delkey` und `passwd` finden können. `adduid` und `deluid` dienen hierbei dem Hinzufügen und Löschen einzelner User-Kennungen. Mit `addkey` und `delkey` können Sie Ihre Teilschlüssel verwalten, also beispielsweise die alle sechs Monate gewechselten Schlüssel für privaten E-Mail-Verkehr. `passwd` schließlich dient dazu, das Mantra zu ändern.

15.13. Ändern der Vertrauensparameter für einen öffentlichen Schlüssel

Manchmal müssen die Vertrauens-Einstellungen für einen öffentlichen Schlüssel geändert werden. Was diese Vertrauensparameter sind, steht im Abschnitt 7.3 auf Seite 63. Mit folgendem Befehl können die Vertrauensparameter für einen öffentlichen Schlüssel geändert werden:

2.6.x: `pgp -ke Benutzer-ID [Schlüsselbund]`
5.0: `pgpk -e Benutzer-ID`
gpg: `gpg --edit-key Benutzer-ID`
Menü: `trust`

Wenn der optionale Parameter `Schlüsselbund` angegeben wird, so muß es ein Bund mit öffentlichen Schlüsseln sein, nicht mit geheimen Schlüsseln.

15.14. Prüfen, ob der Bund mit öffentlichen Schlüsseln intakt ist

Normalerweise prüft PGP automatisch jeden Schlüssel und jede Unterschrift, die einem Bund mit öffentlichen Schlüsseln hinzugefügt werden, und paßt automatisch die Vertrauenseinstellungen und Gültigkeitswerte an. Theoretisch paßt es die Gültigkeitswerte aller betroffenen Schlüssel an, wenn ein Schlüssel dem Bund mit öffentlichen Schlüsseln hinzugefügt oder aus ihm gelöscht wird. Manchmal möchte man aber auch dann eine umfassende Analyse haben, wenn keine Änderungen an dem Schlüsselbund erforderlich sind: Prüfen der Bestätigung(en) der einzelnen Schlüssel, Prüfen der Vertrauensparameter, Neuberechnung der Gültigkeitswerte und Vergleich des eigenen, absolut vertrauenswürdigen Schlüssels mit der Sicherheitskopie auf einer schreibgeschützten Diskette. Es ist sinnvoll, diese Integritäts- und Konsistenzprüfung in regelmäßigen Abständen durchzuführen, um sicherzustellen, daß der Bund mit öffentlichen Schlüsseln wirklich intakt ist. Mit der Option `-kc` führt PGP eine vollständige Analyse des Bundes mit öffentlichen Schlüsseln aus:

```
2.6.x: pgp -kc
  5.0: pgpk -c
  gpg: gpg --check-sigs
```

Mit folgendem Befehl prüft PGP die Unterschriften für einen bestimmten öffentlichen Schlüssel:

```
2.6.x: pgp -kc Benutzer-ID [Schlüsselbund]
  5.0: pgpk -c Benutzer-ID
  gpg: gpg --check-sigs Benutzer-ID
```

Weitere Informationen darüber, wie der eigene Schlüssel mit einer Sicherheitskopie verglichen wird, stehen bei der Beschreibung des Parameters `BAKRING` im Abschnitt 14 auf Seite 106.

Eine komplette Überprüfung des Schlüsselbunds kann auch mit dem Befehl `pgp -km` durchgeführt werden, hierbei zeigt PGP auch die Vertrauensketten an.

15.15. Telephonische Kontrolle eines öffentlichen Schlüssels

Es kann vorkommen, daß man einen öffentlichen Schlüssel bekommt, der nicht von einer anderen vertrauenswürdigen Person bestätigt ist. Dann stellt sich die Frage, wie sich die Echtheit dieses Schlüssels feststellen läßt. Der beste Weg hierfür ist die Kontrolle des Schlüssels über einen anderen Kanal als den, über den der Schlüssel geschickt wurde.

Wenn man die Person kennt, der der öffentliche Schlüssel gehört, und *wenn man auch ihre Stimme am Telephon erkennt,* besteht eine einfache und bequeme Lösung des Problems darin, sie anzurufen, und den Schlüssel im Gespräch zu kontrollieren. Hierzu braucht man sich nicht mühsam den ganzen ASCII-dargestellten Schlüssel vorzulesen. Es reicht, den verhältnismäßig kurzen Fingerabdruck des Schlüssels zu vergleichen. Den Fingerabdruck eines Schlüssel gibt PGP mit der Option `-kvc` aus:

```
2.6.x: pgp -kvc Benutzer-ID [schlüsselbund]
  5.0: pgpk -ll Benutzer-ID
  gpg: gpg --fingerprint Benutzer-ID
```

PGP zeigt die Benutzer-ID zusammen mit dem Fingerabdruck an. Wenn beide Gesprächspartnerinnen diesen PGP-Befehl ausführen, können sie den Schlüssel anhand des Fingerabdruck kontrollieren.

Wenn die Echtheit sowohl des eigenen öffentlichen Schlüssels als auch des öffentlichen Schlüssels des Gesprächspartners überprüft ist, kann die Echtheit der Schlüssel wechselseitig durch eine Unterschrift bestätigt werden. Dies ist ein sicherer und komfortabler Weg, um mit dem Aufbau eines Netzes vertrauenswürdiger Schlüssel innerhalb eines Kreises von Freunden und Freundinnen zu beginnen.

Übrigens bietet sich der Fingerabdruck des eigenen Schlüssels geradezu dazu an, auf die Visitenkarte gedruckt zu werden.

15.16. Ein Wort zu großen öffentlichen Schlüsselbunden

PGP ist ursprünglich entwickelt worden, um kleine Schlüsselbunde mit ein paar hundert Schlüsseln zu verwalten. Mit der Zeit ist PGP aber sehr beliebt geworden, so daß viele Leute nun riesige Schlüsselbunde verwenden. Hierfür ist PGP 2.6.x nicht ausgelegt. Für die Aufnahme einiger

Tausend Schlüssel in Ihren persönlichen Schlüsselbund kann PGP erhebliche Zeit brauchen.

Vielleicht möchten Sie einen riesigen „importierten“ Schlüsselbund Ihrem eigenen hinzufügen, obwohl Sie eigentlich nur an ein paar Dutzend Schlüsseln interessiert sind. Wenn das alles ist, was Sie mit so einem riesigen Schlüsselbund anstellen möchten, gibt es einen effizienteren Weg: Extrahieren Sie die wenigen für Sie interessanten Schlüssel aus dem großen Bund und fügen Sie diese Schlüssel Ihrem eigenen Bund hinzu.

GnuPG können Sie anweisen, einen Schlüsselbund mit richtigen Datenbankmethoden zu verwalten, indem Sie ihn in der `config`-Datei mit `gnupg-gdbm:` vor dem eigentlichen Dateinamen angeben, also beispielsweise

```
keyring gnupg-gdbm:bigring
```

15.17. PGP als Filterprogramm im Unix-Stil

Unix-Fans sind es gewöhnt, „pipes“ zu verwenden, um Daten zwischen zwei Programmen auszutauschen. Der Output eines Programms kann mittels einer „pipe“ als Input in ein zweites Programm gelenkt werden. Damit dies funktioniert, müssen die Programme als „Filter“ arbeiten, also ihre Daten aus dem „standard input“ lesen und auf den „standard output“ schreiben. PGP kann in dieser Form arbeiten.

Falls Ihnen schleierhaft ist, was obiges zu bedeuten hat, werden Sie diese Möglichkeit wahrscheinlich auch nicht brauchen, so daß Sie diesen Abschnitt überspringen können.

Wenn die Option `-f` verwendet wird, arbeitet PGP als Filter. Beispiel:

```
2.6.x: ls -al | pgp -feast Empfänger-ID \
         | mail -s '** Kein Betreff **' emp@faenger.de
  5.0: ps ax | pgpe -fast -r Empfänger \
         | mail -s '** Kein Betreff **' emp@faenger.de
  gpg: cat signed-file | gpg | wc -l
  gpg: echo "hallo" | gpg -ear tom \
         | mail t.budewig@bionic.zerberus.de
```

Diese Option kann die Verwendung von PGP zusammen mit E-Mail-Programmen vereinfachen.

Wenn man PGP als Unix-Filter verwendet, möglicherweise in einem Unix-Skript oder in einer MS-DOS-Batchdatei, kann es sinnvoll sein, die

Umgebungsvariable PGPPASS zu verwenden, um zu verhindern, daß PGP bei jedem Aufruf das Mantra abfragt. PGPPASS ist weiter unten beschrieben.

15.18. Unterdrückung nicht notwendiger Fragen: BATCHMODE

Wird in der PGP-Befehlszeile +batchmode angegeben, stellt PGP während des Programmablaufes keinerlei unnötige Fragen. Bei PGP 5.0 heißt die entsprechende Option -z, bei GnuPG --batch. Beispiel:

```
2.6.x: pgp +batchmode verschlüsselte_datei
  5.0: pgpv -z verschlüsselte_datei
  gpg: gpg --batch verschlüsselte_datei
```

Dies ist sinnvoll, wenn PGP aus Unix-Shell-Skripten oder aus einer MS-DOS-Batchdatei aufgerufen wird. Manche PGP-Befehle, insbesondere für die Schlüsselverwaltung, benötigen in jedem Fall Eingaben durch die Benutzerin. Sie sollten deshalb in Shell-Skripten vermieden werden.

BATCHMODE kann auch bei der Prüfung der Echtheit einer Unterschrift verwendet werden. Ist die Unterschrift nicht in Ordnung, wird als Beendigungscode 1 zurückgegeben. Bei einer intakten Unterschrift gibt PGP den Wert 0 zurück. Bei GnuPG ist hierfür kein gesonderter Parameter notwendig, gpg --verify liefert bereits 1 oder 0 als Rückgabewert.

15.19. „Ja“ oder „Nein“ als Standardantwort bei Bestätigungsabfragen: FORCE/yes/no

FORCE veranlaßt PGP, „ja“ als Standardantwort anzunehmen, wenn es danach fragt, ob eine bereits existierende Datei überschrieben werden soll, oder ob ein öffentlicher Schlüssel aus einem Schlüsselbund mit -kr entfernt werden soll. GnuPG bietet als analoge Option --yes und komplementär --no. PGP 5.0 hat keine entsprechende Option. Beispiel:

```
2.6.x: pgp +force verschlüsselte_datei
  gpg: gpg --yes verschlüsselte_datei
  gpg: gpg --no verschlüsselte_datei
```

oder:

```
2.6.x: pgp -kr +force smith
 gpg: gpg --import --no keyring.asc
```

FORCE kann praktisch sein, wenn PGP aus einem Skript (MS-DOS: Batchdatei) aufgerufen wird.

15.20. Der Beendigungscode von PGP

Um den Betrieb von PGP aus Unix-Skripten oder MS-DOS-Stapeldateien zu unterstützen, gibt PGP eine Statusmeldung an die aufrufende Shell zurück. Ein Beendigungscode von 0 bedeutet, daß kein Fehler auftrat; jeder andere Wert signalisiert einen Fehler. Hiermit können Sie beispielsweise Kommandosequenzen wie diese verwenden:

```
pgp -seat $mailfile $empfaenger
if [ $? -eq 0 ]; then
 mail -s 'encrypted file' $empfaenger <$mailfile.asc
else
 ...
fi
```

Eine Liste der möglichen Fehlercodes, wie sie PGP 2.6.3i liefert, finden Sie in der folgenden Tabelle:

Wert	Bedeutung
0	EXIT_OK kein Fehler
1	INVALID_FILE_ERROR Wird geliefert, wenn beispielsweise bei der Option +makerandom das Schreiben einer Datei fehlgeschlagen ist (Platte voll? Quota?), wenn ein zu langer Dateiname eingegeben wurde, wenn Sie versuchen, eine Binärdatei mit -m zu verschlüsseln, auf dem MacIntosh, wenn die Konvertierung einer Mac-Datei in das portable Binärformat fehlschlägt, wenn die ASCII-Transporthülle einer Datei beschädigt wurde, wenn PGPs Versuche, die Eigenschaften Ihres Terminals zu erfragen oder zu setzen, fehlschlagen oder PGP es nicht schafft, einzelne Zeichen direkt zu lesen oder – das ist die einzige Nicht-Fehlerbedingung, unter der dieser Rückgabewert auftritt – beim Prüfen einer Signatur mit +batchmode, wenn keine gültige Signatur gefunden wurde.

Wert	Bedeutung
2	FILE_NOT_FOUND_ERROR Eine Eingabedatei wurde nicht gefunden.
3	UNKNOWN_FILE_ERROR Ein allgemeiner Dateifehler; die Datei konnte nicht überschrieben werden (-w), das Schreiben auf „stdout“ oder allgemein einer Ausgabe oder Temporärdatei ist fehlgeschlagen oder das Datenformat einer Eingabedatei ist korrupt oder kein PGP-Format.
4	NO_BATCH Es wurde +batchmode bei einem Befehl gesetzt, wo das nicht möglich ist (Schlüsselerzeugung, Hinzufügen eines User-ID, Ändern des Mantras, Schlüssel zurückziehen/sperren, Löschen von Unterschriften).
5	BAD_ARG_ERROR In config.txt oder auf der Befehlszeile sind unbekannte Parameter verwendet worden. pgp -h liefert auch diesen Rückgabewert.
6	INTERRUPT Das Programm wurde unterbrochen.
7	OUT_OF_MEM Es steht nicht genügend Speicher zur Verfügung.
10	KEYGEN_ERROR Beim Erzeugen eines Schlüssels ist ein Fehler aufgetreten. Beachten Sie hierzu bitte auch Abschnitt 14 auf Seite 122.
11	NONEXIST_KEY_ERROR pgp -ka wurde mit dem Namen einer nicht existenten Datei als Schlüsselbund aufgerufen.
12	KEYRING_ADD_ERROR Beim Hinzufügen eines Schlüssels zum Schlüsselbund ist ein Fehler aufgetreten.
13	KEYRING_EXTRACT_ERROR Beim Extrahieren eines Schlüssels konnte die Datei mit dem Schlüsselbund nicht geöffnet werden oder sie ist kaputt, das Extrahieren wurde im batchmode aufgerufen, ohne eine Ausgabedatei anzugeben oder es wurde kein Ausgabedateiname eingegeben (direkt Return gedrückt), der Schlüssel ist in der Ausgabedatei bereits enthalten oder der User hat angegeben, die Ausgabedatei solle nicht überschrieben werden oder sie ist nicht schreibbar.
14	KEYRING_EDIT_ERROR Allgemeiner Fehler bei pgp -ke.
15	KEYRING_VIEW_ERROR Fehler in einer Datei mit Schlüsseln, entweder im Schlüsselbund selbst oder einer externen Datei. Vermutlich ist die betreffende Datei defekt.
16	KEYRING_REMOVE_ERROR Das Löschen eines Schlüssels aus dem Bund ist fehlgeschlagen, beispielsweise durch Abbruch durch den Benutzer.

Wert	Bedeutung
17	KEYRING_CHECK_ERROR Bei pgp -kc oder pgp -km sind interne Inkonsistenzen gefunden worden.
18	KEY_SIGNATURE_ERROR Beim Unterschreiben eines Schlüssels ist ein Fehler aufgetreten, beispielsweise Abbruch durch die Anwenderin.
19	KEYSIG_REMOVE_ERROR Das Löschen einer Unterschrift unter einem Schlüssel ist fehlgeschlagen, beispielsweise durch Abbruch durch den Benutzer.
20	SIGNATURE_ERROR Fehler beim Unterschreiben einer Datei oder Nachricht, beispielsweise falsches Mantra.
21	RSA_ENCR_ERROR Fehler bei der asymmetrischen Verschlüsselung, vermutlich wurde der Schlüssel des Empfängers nicht gefunden.
22	ENCR_ERROR Fehler bei der konventionellen Verschlüsselung, beispielsweise der Versuch, ohne Mantra zu verschlüsseln.
23	COMPRESS_ERROR Dieser Fehler ist definiert, wird aber nirgends ausgelöst.
30	SIGNATURE_CHECK_ERROR Beim Prüfen einer Unterschrift ist ein Fehler aufgetreten. Dieser Fehler wird bislang nicht ausgelöst.
31	RSA_DECR_ERROR Bei der Entschlüsselung mit Hilfe des privaten Schlüssels ist ein Fehler aufgetreten. Vermutlich ist die Datei nicht für Sie verschlüsselt, evtl. auch defekt.
32	DECR_ERROR Bei der Entschlüsselung einer konventionell verschlüsselten Datei ist ein Fehler aufgetreten. Vermutlich ist das eingegebene Mantra falsch.
33	DECOMPRESS_ERROR Beim Dekomprimieren ist ein Fehler aufgetreten. Vermutlich ist die Datei defekt.

15.21. Umgebungsvariable für das Mantra: PGPPASS

Normalerweise fragt PGP das Mantra genau zu dem Zeitpunkt ab, zu dem ein geheimer Schlüssel benötigt wird. Das Mantra kann aber auch in der Umgebungsvariablen PGPPASS gespeichert werden. Wenn PGPPASS definiert ist, versucht PGP, ihren Wert als Mantra für den Zugriff auf einen geheimen Schlüssel zu verwenden. Ist das in PGPPASS gespeicherte Mantra nicht korrekt, fragt PGP nach dem richtigen Mantra.

Unter MS-DOS könnte das Mantra so gesetzt werden:

```
SET PGPPASS=Zaphod Beeblebrox wird Bundespräsident
```

Die Eingabe eines Mantra während des Laufs von PGP ist dann nicht mehr erforderlich, vorausgesetzt, daß „Zaphod Beeblebrox wird Bundespräsident“ wirklich das richtige Mantra ist.

Nein, das wäre kein gutes Mantra, insbesondere, wenn Sie Douglas Adams-Fan sind. Nicht einmal, wenn wir es nicht abgedruckt hätten. „Am gelben Auto ein 25er Diamant.“ oder so etwas ist schon etwas besser. Was wirklich gut ist, hängt davon ab, wie gut Sie es sich merken können — Ihr Mantra sollte so kompliziert und so wenig zu Ihnen passend sein, daß Sie es gerade noch behalten und schnell tippen können,+ ohne daß Sie es aufschreiben müssen.◁ Wenn Sie sich etwas wie `tte/RU66WcgIa.FO};6DeAy??zu@7dzc-iIsP+` merken können und bereit sind, es täglich mehrmals zu tippen, wäre das natürlich toll und sehr, sehr sicher. Außerdem hätten Sie dann meine Hochachtung – ich könnte mir das nie merken.

Die Verwendung von PGPPASS ist einerseits riskant, macht aber andererseits die Arbeit mit PGP wesentlich einfacher, wenn man regelmäßig größere Mengen verschlüsselter Nachrichten erhält.

Die Auswertung der Umgebungsvariablen PGPPASS hat PHILIP ZIMMERMANN auf vielfachen Wunsch hin in PGP aufgenommen. Die Verwendung von PGPPASS ist aber gefährlich, weil das Mantra dann nicht nur in Ihrem Gedächtnis, sondern auch irgendwo im Speicher Ihres Rechners steht. Leichtsinnig wäre es, das Mantra in einer Datei auf demselben Rechner zu speichern, auf dem auch Ihre Datei `secring.pgp` steht. Nicht nur sehr leichtsinnig, sondern einfach dumm wäre es, das Mantra in einer Batchdatei, am Ende gar in `autoexec.bat`, zu speichern. Jedermann könnte sich in der Mittagspause an Ihren Rechner setzen und sowohl Ihr Mantra als auch Ihren geheimen Schlüsselbund mitgehen lassen.

Die Gefährlichkeit von PGPPASS kann nicht stark genug betont werden.

Falls Sie PGPPASS unbedingt brauchen, ist es am sichersten, das Kommando zum Setzen von PGPPASS unmittelbar vor der Benutzung von PGP einzutippen und unmittelbar nach der Benutzung von PGP PGP–

+ Wenn Sie es langsam tippen müssen, kann jemand, der die Tastatur sehen kann, Ihr Mantra gewissermaßen „mitlesen“.

◁ Das Mantra aufzuschreiben, ist fast genauso unvorsichtig, wie keines zu verwenden. Was Sie sich aufschreiben, können auch andere lesen.

PASS wieder zu löschen oder den Computer auszuschalten. Sie sollten PGPPASS *auf keinen Fall* verwenden, wenn andere Personen Zugang zu Ihrem Computer haben. Jemand könnte vorbeikommen und Ihr Mantra ganz einfach durch Abfragen der Umgebungsvariablen herausfinden.

PGP stellt noch weitere, weniger bedenkliche Methoden zur Verfügung, mit denen die Eingabe des Mantras automatisiert werden kann. Auch diese sind Sicherheitsrisiken, aber kleinere. Das Mantra kann mit der Option -z in der Befehlszeile übergeben werden:

2.6.x: `pgp -m "-zZaphod Beeblebrox wird Kanzler" geheim.pgp`

Oder es wird in eine (vor anderen Benutzern geschützte) Datei geschrieben und der Pfad und der Dateiname dieser Datei in der Umgebungsvariablen PGPPASSFD gespeichert.

GnuPG bietet etwas ähnliches, hier muß aber ein „file descriptor“ angegeben werden. Die Details sind eher unwichtig und meistens wird die Funktionalität vermutlich verwendet, wenn GnuPG von einem anderen Programm aus gestartet wird. Wenn Sie die bash verwenden, funktioniert z. B. so ein Aufruf:

gpg: `gpg --passphrase-fd 42 -sb datei.txt 42<~/.gpg/mantra`

Bedenken Sie bei allen diesen „Vereinfachungen“, daß Systemadministratoren auf praktisch allen Systemen problemlos alle Dateien und Umgebungsvariablen lesen können. Es ist natürlich richtig, daß sie auch laufenden Prozessen „über die Schulter schauen“ können, aber das ist wesentlich aufwendiger und bedarf eines sehr viel genaueren Timings. Sie sollten auch nicht vergessen, daß die eigentlichen Systemadministratoren oft überrascht wären, wer alles Administrator-Rechte auf ihrem System hat. Außerdem hat der Aufruf mit der Option -z den Nachteil, daß Ihr Mantra dann auf der Kommandozeile steht. Zum Einen gelangt es so in die „history“ Ihrer Shell (unter MS-DOS reicht sie im Original nur eine Zeile weit, aber mit 4DOS sieht das schon wieder anders aus . . .), zum anderen ist die Kommandozeile eines laufenden Prozesses auch für andere Benutzer sichtbar. PGP überschreibt den entsprechenden Teil zwar mit Leerzeichen, aber in der Zeit bis dahin ist Ihr Mantra für jeden eingeloggten Benutzer lesbar.

Teil III.

Windowsversionen

16. Allgemeines

16.1. Vorbemerkungen

Dieser dritte Teil des Handbuches beinhaltet die Installations- und Bedienungsanleitung für die PGP-Versionen mit graphischer Benutzeroberfläche unter Microsoft Windows 9x und Windows NT 4.0.

Es geht dabei vor allem um die Bedienung, die sich sehr stark von den Kommandozeilenversionen unterscheidet, die im Teil II dieses Buches behandelt wurden. Für das Verständnis der grundlegenden Funktions- und Arbeitsweisen von PGP lesen Sie bitte im Teil I dieses Buches nach. Sie werden in diesem Teil z. B. keine Erklärungen darüber finden, was das Signieren einer Nachricht bedeutet, sondern nur darüber, wie Sie dies in den PGP-Versionen mit graphischer Oberfläche ausführen können. Das grundsätzliche Verständnis von asymmetrischer Verschlüsselung und der entsprechenden Begriffe wird hierbei vorausgesetzt.

Anders gesagt: Es ist absolut *sinnlos,* dieses Kapitel zu lesen, ohne wenigstens die in der Einleitung dringend empfohlenen Kapitel studiert zu haben. Außerdem setzen wir voraus, daß Sie mit der Bedienung Ihres Betriebssystems (Microsoft Windows 95, 98 oder NT 4.0) vertraut sind.

Die Dialogboxen in dieser Anleitung stammen teilweise aus einem US-englischen Windows, wenn Sie ein deutsches Windows installiert haben, werden die Beschriftungen der Schaltflächen (Buttons) anders lauten („Ja“ statt „Yes“ usw.).

Die Version 6.5.1i ist kurz vor Drucklegung dieses Buchs erschienen. Die kurze Zeit hat es uns leider nicht erlaubt, in der gebotenen Ausführlichkeit auf sie einzugehen. Die Installation unterscheidet sich nur in sehr wenigen Punkten von der der Version 6.0i, deshalb gibt es kein eigenes Kapitel für die Installation der Version 6.5.1i. Die Schlüsselerzeugung wiederum entspricht der Version 5.5.3i am ehesten, denn 6.5.1i gestattet es Ihnen, RSA-Schlüssel zu erzeugen. Die in 6.5.1i neu hinzugekommene Funktionalität wird in diesem Handbuch nicht besprochen.

16.2. Systemvoraussetzungen

Alle in diesem Buch behandelten Versionen von PGP für Windows (5.0i, 5.53i, 6.0i und 6.5i) sind 32-Bit-Programme für Microsoft Windows 95, Windows 98 oder Windows NT. Sie sind allesamt nicht lauffähig auf Systemen mit Microsoft MS-DOS/Windows 3.x als Betriebssystem.[⊤] Aus unbestätigten Nachrichten haben wir entnommen, daß sie sich auch nicht unter Windows 2000 installieren lassen; das dürfte aber „lediglich“ ein Problem mit den Installationsroutinen sein.

Wenn Sie noch mit einem Rechner unter MS-DOS und Microsoft Windows 3.x als Betriebssystem arbeiten und darauf PGP einsetzen möchten, dann müssen Sie eine Kommandozeilenversion einsetzen, gegebenenfalls mit einem graphischen Zusatzprogramm zur Bedienung. Näheres finden Sie in Teil II auf Seite 68.

Aber auch für Benutzerinnen von Windows 95/98/NT gibt es teilweise gute Gründe, die älteren PGP-Versionen 2.6.xi einzusetzen (z. B. der in mancher Hinsicht eingeschränkte Funktionsumfang der PGP-Versionen für Windows). Bitte lesen Sie hierzu die entsprechenden Abschnitte in Abschnitt B auf Seite 271. Für die Benutzung von PGP 2.6.xi unter den Windows-Versionen 3.1, 3.11, 95, 98 und NT gibt es zusätzlich zur Kommandozeile die Möglichkeit, PGP über eine graphische Benutzerschnittstelle fernzusteuern. Wir haben entsprechende Programme auf der beiliegenden CD im Verzeichnis `Frontends` versammelt.

Ein Beispiel für eine solche graphische Benutzerschnittstelle ist das Shareware-Programm `PGPClick` von ROBERT ELDEN WILSON, das auch auf der beiliegenden CD enthalten ist. Bitte beachten Sie, daß es sich um Shareware und nicht um kostenlose Freeware handelt und registrieren Sie das Programm, wenn Sie es benutzen möchten. Es gibt Versionen für Windows 3.1x, aber auch für Windows 95/98/NT, falls Sie die ältere Version von PGP verwenden möchten, aber nicht auf eine graphische Oberfläche verzichten möchten oder können. Die Software ist auf Basis von Visual Basic 4.0 geschrieben, daher müssen die Visual Basic 4.0 Runtime-Bibliotheken auf dem Rechner installiert werden, damit das Programm läuft. Sie finden diese Bibliotheken ebenfalls auf der beiliegenden CD. Nähere Informationen zu `PGPClick` finden Sie unter `http://www.ncinter.net/~rewilson/PGPClick.html`.

[⊤] Es gibt eine MS-DOS-Version von PGP 5.0i. Sie finden ihre Besprechung in Teil II ab Seite 68.

17. Installation

Die verschiedenen Versionen von PGP für Windows unterscheiden sich teilweise erheblich, was das Verhalten während der Installation angeht. Um die Übersichtlichkeit der Anweisung nicht allzu sehr zu strapazieren, haben wir die Installationsanleitung für die unterschiedlichen Versionen getrennt aufgeführt.

17.1. Vorsicht, Falle – was Sie bei der Installation aller Versionen von PGP für Windows beachten sollten

Alle Versionen von PGP für Windows schlagen Ihnen im Laufe der Installation vor, das Programm für die Schlüsselverwaltung, `PGPkeys`, nach Abschluß der Installation automatisch zu starten. Vom Grundgedanken her ist das sicher auch sinnvoll, denn schließlich benötigt man zuerst einmal Schlüssel, um PGP benutzen zu können.

Die Angelegenheit hat aber einen Pferdefuß: PGP verwendet dann Grundeinstellungen, die unserer Meinung nach dazu führen, daß die Sicherheit der Schlüssel beeinträchtigt wird, wenn auch nur gering.

PGP für Windows (alle in diesem Buch vorgestellten Versionen) verwendet in der Standardeinstellung für die Berechnung von ElGamal-Schlüsseln eine Auswahl vorausberechneter großer Primzahlen.• Die Berechnung solcher Primzahlen ist nämlich recht aufwendig und die Schlüsselerzeugung dauert ohne diese „Abkürzung“ um ein vielfaches länger, als wenn die vorausberechneten Primzahlen verwendet werden. Für einen einigermaßen paranoiden Einsatz (gemäßigte Paranoia dient der Sicherheit) empfiehlt es sich, diese Option abzuschalten. Der Zeitgewinn durch die Verwendung der (allgemein bekannten) eingebauten Primzahlen wird ausschließlich bei der Schlüsselerzeugung erreicht, bei der Schlüsselverwendung (also bei der alltäglichen Benutzung) wird

• Der „Modulus“ des öffentlichen Schlüssels ist eine Primzahl q. ElGamal ist unsicher, wenn $(q-1)/2$ kleine Teiler hat. Also versucht PGP, ein q der Form $2*p_1*p_2*\cdots*p_n+1$ zu finden. Näheres steht in Abschnitt D.3 auf Seite 279.

keine Zeit gespart. Da neue Schlüssel nicht ständig erzeugt werden müssen, sondern einmal erzeugte Schlüssel über einen längeren Zeitraum genutzt werden, sollten Sie sich die Zeit nehmen, das Programm neue Primzahlen berechnen zu lassen.

Die Einstellung betrifft nur DSS/ElGamal-Schlüssel. Für die in den älteren Versionen ausschließlich benutzten RSA-Schlüssel hat sie keine Funktion (diese können aber in den Versionen 5.0i und 6.0i sowieso nicht erzeugt werden).

Abbildung 17.1: `PGPkeys` starten? (5.0i)

Wenn Sie `PGPkeys` nach Abschluß der Installation automatisch starten lassen (dies ist bei allen Installationsprogrammen die Voreinstellung), startet `PGPkeys` mit den Standardeinstellungen, die auch diese schnelle Schlüsselerzeugung beinhalten. Da beim ersten Start des Programms noch kein Schlüssel vorhanden ist, wird automatisch das Programm für die Schlüsselerzeugung aufgerufen. In diesem Fall würde Ihr neuer Schlüssel mit der Option der schnellen Schlüsselerzeugung, also vorgegebenen Primzahlen, erzeugt.

Wir empfehlen Ihnen daher, zunächst die Option `Faster Key Generation` in den PGP-Grundeinstellungen abzuschalten, bevor Sie `PGPkeys` das erste Mal starten. Hierzu sollten Sie die Option zum Starten von `PGPkeys` am Ende der Programminstallation abwählen (Abb. 17.1).

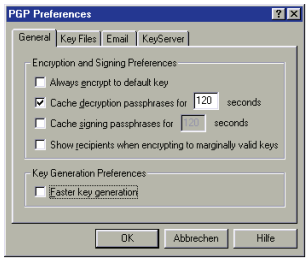


Abbildung 17.2: Eigene Primzahlen suchen (5.0i)

Nach Abschluß der Installation rufen Sie die PGP-Grundeinstellungen auf, indem Sie mit der Maus auf das `PGPtray`-Symbol klicken (den Briefumschlag mit Vorhängeschloß in der rechten Ecke Ihrer Windows-Taskleiste). Daraufhin erscheinent ein Menü. In diesem wählen Sie den Punkt `PGP Preferences` aus. In dem dann erscheinenden Fenster wählen Sie die Option `Faster Key Generation` durch Klicken auf das Häkchen ab (Abb. 17.2) und bestätigen die Änderung mit `OK`.

Nun können Sie `PGPkeys` entweder über die entsprechenden Einträge im Windows-Startmenü oder aus dem Menü von `PGPtray` starten. Die Anleitung zum Erzeugen von Schlüsseln finden Sie in Abschnitt 18.1 auf Seite 166.

Sollten Sie das Abwählen des automatischen Starts von PGPkeys im Zuge der Installation vergessen haben, können Sie die Erzeugung eines neuen Schlüssels auch nach dem Start des Schlüsselerzeugungsprogramms Key Generation Wizard durch Klicken auf die Schaltfläche Abbrechen vorzeitig beenden. PGPkeys startet dann ohne Schlüssel. Sie können nach dem Start des Programms die PGP-Grundeinstellungen ändern (wie oben beschrieben oder über den Menüpunkt Edit/Preferences in PGPkeys) und anschließend einen neuen Schlüssel durch Aufruf des Menüpunkts Keys/New Key in PGPkeys erzeugen.

17.2. Installation von PGP Freeware 5.0i für Windows

PGP Freeware 5.0i für Windows 95, Windows 98 oder Windows NT 4.0 kommt in Form einer Zip-komprimierten Datei PGP50i-win95nt.zip, die Sie auch auf der beiliegenden CD finden. Diese Datei muß für die Installation zuerst mit einer entsprechenden Entpacker-Software,▽ die Zip-Archive dekomprimieren kann (z. B. der Freeware Alladin Expander), in ein beliebiges (am besten leeres) Verzeichnis auf der Festplatte ausgepackt werden. Eine Diskette reicht aus Platzgründen hierfür nicht aus. Die vom Entpacker-Programm erzeugten Dateien werden nur für die Installation benötigt und können nach erfolgreicher Installation wieder gelöscht werden.

Näheres zum Dekomprimieren des Zip-Archivs entnehmen Sie bitte der Dokumentation des jeweiligen von Ihnen benutzten Programmes. Für Alladin Expander genügt es, das Archiv im Explorer mit einem Doppelklick anzuwählen. Ist das aktuelle Verzeichnis schreibbar, wird das Archiv in ein neues Verzeichnis im aktuellen Verzeichnis entpackt; ansonsten werden Sie aufgefordert, ein Zielverzeichnis auszuwählen.

Nach dem Auspacken finden sich im Zielverzeichnis folgende Dateien und Verzeichnisse:

Keygen.avi Eine Animation, die bei der Schlüsselerzeugung angezeigt wird.

License.txt Der Benutzerlizenzvertrag – die Benutzung von PGP 5.0i ist nur für den nicht-kommerziellen Gebrauch kostenlos!

PGP50.hlp Die Windows-Hilfe-Datei zum Programm.

▽ Die Entpacker-Software muß mit langen Dateinamen umgehen können.

PGP50manual.pdf Das Original-Handbuch zum Programm in Form einer Adobe PDF Datei. Um das Handbuch lesen zu können, benötigen Sie ein Programm, das Adobe PDF-Dateien darstellen kann, z. B. die auf der CD beiliegende Freeware Adobe Acrobat Reader. Dieses Programm müssen Sie gesondert installieren, es ist in PGP nicht enthalten. Bitte lesen Sie hierzu die Dokumentation des Programmes, das Sie für die Anzeige der PDF-Dateien verwenden möchten.

PGPkeys.exe Das Programm zur Verwaltung von Schlüsseln.

PGPtray.exe Ein Programm zum einfachen Aufrufen der verschiedenen PGP-Funktionen. Es richtet sich im System-Tray der Windows-Startleiste ein.

Readme.txt Ein Text mit Informationen über das Programm sowie mit den letzten Änderungen, die in der übrigen Dokumentation eventuell nicht mehr berücksichtigt werden konnten.

Setup.exe Das Installationsprogramm.

Welcome.txt Ein Begrüßungstext mit Informationen über die Entstehungsgeschichte dieser Version und Hinweisen auf einige (im Vergleich zu 2.6.x) neue Programmfunktionen.

Sig Ein Verzeichnis, in dem sich für die beim Auspacken erzeugten Dateien abgetrennte Signaturen der Entwickler befinden, die mit (einem bereits installierten, vertrauenswürdigen) PGP überprüft werden können.

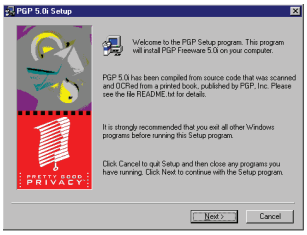


Abbildung 17.3: Der Begrüßungsschirm (5.0i)

Die Installation wird durch Ausführen von `Setup.exe` gestartet (z. B. durch einen Doppelklick auf das Programmsymbol). Als erstes erscheint ein Begrüßungsbildschirm (Abb. 17.3), der Ihnen (wie beim Installieren neuer Programme üblich) empfiehlt, alle noch laufenden Programme außer dem Installationsprogramm zu beenden. Wenn Sie dies getan haben, bestätigen Sie mit einem Mausklick auf `Next`.

Als zweiter Bildschirm wird die Benutzerlizenz angezeigt und gefragt, ob Sie den darin beschriebenen Bedingungen zustimmen. Wenn Sie die

Vereinbarung gelesen haben und ihr zustimmen, bestätigen Sie dies bitte durch Anwahl des Knopfes `Yes`. Wenn Sie den Bedingungen nicht zustimmen und auf `No` klicken, beendet sich das Installationsprogramm. PGP wird dann nicht installiert.

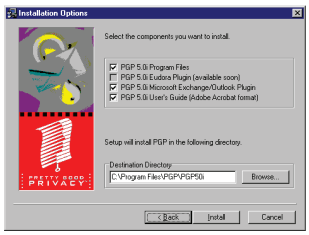


Abbildung 17.4: Auswahl der Komponenten (5.0i)

Im dritten Bildschirm geben Sie an, welche Komponenten das Programm auf Ihrem Rechner installiert und in welchem Verzeichnis auf Ihrer Festplatte dies geschehen soll (Abb. 17.4). Alle verfügbaren Komponenten sind standardmäßig schon zur Installation ausgewählt. Durch Klicken auf die Häkchen links von den Komponenten können Sie einzelne Komponenten von der Installation ausnehmen.

Folgende Komponenten werden angeboten:

PGP 5.0i Program Files Die für die Ausführung benötigten Programme. Diese Option sollte in jedem Fall aktiviert bleiben, wenn Sie das Programm installieren möchten. Ein Abschalten dieser Option macht nur Sinn, wenn Sie PGP bereits installiert haben und das Installationsprogramm nutzen möchten, um eine Erweiterung für ein E-Mailprogramm nachzuinstallieren.

PGP 5.0i Eudora Plugin Diese Option wird zwar aufgeführt, kann aber nicht angewählt werden. Die PGP-Erweiterung für das E-Mailprogramm Qualcomm Eudora (light) wurde erst später fertiggestellt, für PGP 5.0i müssen Sie daher dieses Plugin gesondert installieren. Sie finden das Plugin als Zip-komprimiertes Archiv unter dem Dateinamen `PGP50Eud305W95.zip` auf der beiliegenden CD.

PGP 5.0i Microsoft Exchange/Outlook Plugin Wenn Sie als E-Mailprogramm Microsoft Exchange/Outlook verwenden, sollten Sie diese Option aktiviert lassen, damit Ihnen die PGP-Funktionen direkt in Ihrem E-Mailprogramm zur Verfügung stehen. Wenn Sie ein anderes Programm für die Bearbeitung Ihrer E-Mails verwenden, können Sie diese Option abwählen. Beachten Sie bitte, daß auch Microsoft Outlook Express, der mit dem Internet Explorer ausgeliefert wurde, ein anderes Programm als Exchange/Outlook ist und dieses Plugin nicht in Outlook Express funktioniert. Hierfür gibt es

in späteren PGP-Versionen eine gesonderte Erweiterung, die aber mindestens PGP 5.5.3i benötigt.

PGP 5.0i User's Guide (Adobe Acrobat Format) Wenn Sie das elektronische Original-Handbuch installiert haben möchten, sollten Sie diese Option aktiviert lassen. Für die Funktion des Programms ist diese Option nicht nötig. Um das Handbuch lesen zu können, benötigen Sie ein Programm, das Adobe PDF-Dateien darstellen kann, z. B. die auf der CD beiliegende Freeware Adobe Acrobat Reader. Dieses Programm müssen Sie gesondert installieren, es ist in PGP nicht enthalten. Bitte lesen Sie hierzu die Dokumentation des Programmes, das Sie für die Anzeige der PDF-Dateien verwenden möchten.

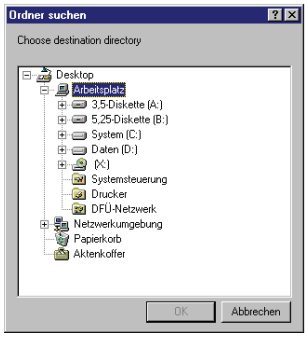


Abbildung 17.5: Installationsverzeichnis wählen (5.0i)

Das Installationsprogramm schlägt Ihnen im Dialogfeld `Destination Directory` ein Verzeichnis vor, in dem PGP installiert werden soll. Dieses Verzeichnis können Sie ändern, indem Sie auf `Browse` klicken und in dem erscheinenden Dialogfeld (Abb. 17.5) ein neues Verzeichnis auswählen, oder Sie können einen neuen Installationspfad direkt in das Feld `Destination Directory` eingeben. Sollte das von Ihnen gewählte Verzeichnis noch nicht existieren, fragt das Installationsprogramm nach, ob es das Verzeichnis erstellen soll. Bitte bestätigen Sie hier mit `Yes`.

Das Installationsprogramm hängt an das von Ihnen im Browse-Fenster ausgewählte Verzeichnis automatisch immer `PGP50i` an. Wenn Sie PGP z. B. in einem Verzeichnis `C:\PGP50i` installieren möchten, genügt es, Laufwerk `C:` im Fenster als Installationsziel anzuklicken. Soll das Zielverzeichnis anders als `PGP50i` heißen, so geht dies nur über die direkte Eingabe des Pfades im Feld `Destination Directory`.

Anmerkung: Das Installationsprogramm schlägt Ihnen den Pfad `C:\Program Files\PGP\PGP50i` vor. Auf deutschsprachigen Windows-Versionen heißt der Ordner `C:\Program Files` jedoch `C:\Programme`. Der Einheitlichkeit halber können Sie den Pfad anpassen, die Installation funktioniert aber auch mit der Standardvorgabe oder mit jedem anderen Verzeichnis. (Sie haben bei Benutzung der Standardvorgabe dann

ein Verzeichnis `Program Files` und ein Verzeichnis `Programme` auf Ihrer Festplatte.)

Mit Anklicken von `Install` starten Sie die eigentliche Installation, mit `Cancel` können Sie die Installation hier noch abbrechen, ohne daß an Ihrem System etwas verändert wurde (nur das Verzeichnis wurde erstellt, wenn es noch nicht existiert hat). Wenn Sie `Install` gewählt haben, beginnt das Installationsprogramm mit dem Kopieren der Programmdateien in das von Ihnen vorgegebene Installationsverzeichnis sowie einiger Programmbibliotheken in das Windows-Verzeichnis.

Nach kurzer Zeit erscheint ein weiterer Bildschirm mit der Meldung, daß die Installation komplett und erfolgreich war (zumindest sollte das so sein; Abb. 17.1). Sie können in diesem Fenster auswählen, ob Sie sich die `Readme`-Datei anzeigen lassen möchten und ob das Programm `PGPkeys` für die Schlüsselverwaltung nach Abschluß der Installation gestartet werden soll. Es empfiehlt sich, hier die Option `PGPkeys` abzuschalten. Bitte beachten Sie dazu den Abschnitt 17.1 am Anfang dieses Kapitels.

Durch Klicken auf `OK` wird die eigentliche Programminstallation beendet. Wenn Sie im letzten Bildschirm die jeweiligen Optionen gewählt hatten, wird nun im Anschluß an die Installation die `Readme`-Datei angezeigt und das Programm `PGPkeys` zur Schlüsselverwaltung gestartet.

Eine Anleitung zur Erzeugung von Schlüsseln finden Sie im Abschnitt 18.1 auf Seite 166.

Installation von `PGPtray`

`PGPtray` ist ein Programm, das Ihnen die Funktionen für die Verschlüsselung und Signatur bzw. für die Entschlüsselung und Signaturprüfung über die Windows-Zwischenablage im Windows-System-Tray (rechte Seite der Windows-Taskleiste) zur Verfügung stellt. Außerdem können Sie darüber andere PGP-Funktionen wie das Ändern der Grundeinstellungen oder die Schlüsselverwaltung `PGPkeys` sehr einfach aufrufen.

Wenn Sie ein E-Mailprogramm benutzen, für das es keine PGP-Erweiterung gibt, müssen Sie die PGP-Verschlüsselung und Signatur bzw. die Entschlüsselung und Signaturprüfung vornehmen, indem Sie die zu bearbeitenden Daten in die Windows-Zwischenablage („Clipboard“) kopieren und dann über `PGPtray` die PGP-Funktionen auf die Daten in der Zwischenanlage anwenden. Um in diesem Fall PGP benutzen zu können, muß `PGPtray` unbedingt gestartet sein.

Das Installationsprogramm von PGP 5.0i richtet zwar den Aufruf

von `PGPtray` im Windows-Startmenü ein, es ist aber für die regelmäßige Benutzung sinnvoll, `PGPtray` gleich beim Systemstart mit starten zu lassen, ansonsten müssen Sie `PGPtray` jedesmal von Hand starten, bevor Sie PGP mit der Windows-Zwischenablage benutzen können. Hierzu sollten Sie im Windows-Startmenü im Ordner `Programme\Autostart` eine Verknüpfung zum Programm `PGPtray` erstellen. Es befindet sich im von Ihnen im Laufe der PGP-Installation ausgewählten Zielverzeichnis. Die neueren Versionen der Installationsprogramme von PGP 5.5.3i und PGP 6.0i erstellen diese Verknüpfung automatisch.

17.3. Installation von PGP Freeware 5.5.3i für Windows

PGP Freeware 5.5.3i für Microsoft Windows 95/98/NT kommt in Form eines ausführbaren Programms `pgp553i-win95nt.exe`, das Sie auch auf der beiliegenden CD finden. Das Programm enthält die Installationsroutine und die benötigten Daten in komprimierter Form. Die Installation wird durch Ausführen des Programms, z. B. durch Doppelklick auf das Programmsymbol, gestartet.

Abbildung 17.6: Der Begrüßungsbildschirm (5.5.3i)

Das (originale, US-englische) Benutzerhandbuch in Form einer Adobe PDF-Datei ist in diesem Installationsprogramm nicht enthalten. Sie müssen das Handbuch als getrennte Datei herunterladen bzw. von der beiliegenden CD in das gewünschte Verzeichnis kopieren. Der Dateiname für das Handbuch lautet `pgp55win.pdf`. Um diese Datei lesen zu können, benötigen Sie ein Programm, das Adobe PDF-Dateien darstellen kann, z. B. die auf der CD beiliegende Freeware Adobe Acrobat Reader. Dieses Programm müssen Sie gesondert installieren, es ist in PGP nicht enthalten. Bitte lesen Sie hierzu die Dokumentation des Programms, das Sie für die Anzeige der PDF-Dateien verwenden möchten.

Nachdem das Installationsprogramm die Daten dekomprimiert hat, erscheint ein Begrüßungsbildschirm (Abb. 17.6), der Ihnen empfiehlt, alle noch laufenden Programme außer dem Installationsprogramm zu beenden. Wenn Sie dies getan haben, bestätigen Sie mit einem Mausklick auf `Next`.

Als nächstes wird der Lizenzvertrag für PGP Freeware 5.5.3i angezeigt,

den Sie für die Installation des Programms mit einem Mausklick auf `Yes` akzeptieren müssen. Sollten Sie ihn mit `No` ablehnen, beendet sich das Installationsprogramm ohne Installation von PGP. PGP Freeware 5.5.3i darf ausschließlich nicht-kommerziell eingesetzt werden!

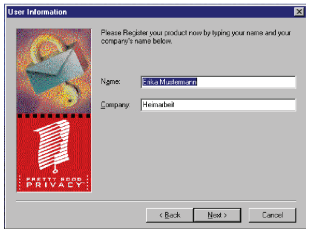


Abbildung 17.7: Eingabe der Benutzerdaten (5.5.3i)

Im nächsten Bildschirm müssen Sie in die dafür vorgesehenen Felder Ihren Namen und einen Firmennamen eingeben (Abb. 17.7). Die Eingabe eines Namens und einer Firma ist obligatorisch,♠ die Installation kann erst fortgesetzt werden, wenn beide Eingabefelder nicht leer sind, ansonsten ist die Schaltfläche `Next` deaktiviert. Das Installationsprogramm übernimmt als Voreinstellung die in Windows gespeicherten Benutzereinstellungen, die überschrieben werden können. Mit `Next` geht es weiter.

Im folgenden Bildschirm werden Sie aufgefordert, ein Zielverzeichnis für die Installation anzugeben (Abb. 17.8). Sie können entweder durch Anklicken von `Next` die Voreinstellung `C:\Program Files\PGP\PGP55i` übernehmen oder nach Anklicken der Schaltfläche `Browse` ein anderes Verzeichnis auswählen. In diesem Fall erscheint ein Fenster mit einem Verzeichnisbaum, wo Sie durch Auswählen aus dem Listenfeld `Drives` (Laufwerke) ein anderes Laufwerk als `C:` wählen können und durch Anklicken der Symbole für die auf Ihrem Rechner vorhandenen Verzeichnisse ein anderes Verzeichnis auswählen können. Bestätigung und Rückkehr zum ursprünglichen Bildschirm mit `OK`.

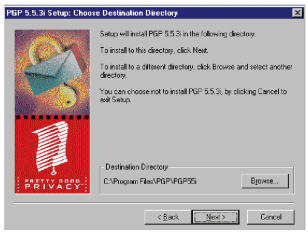


Abbildung 17.8: Auswahl des Installationsverzeichnisses (5.5.3i)

Möchten Sie PGP in einem Verzeichnis installieren, das noch nicht existiert, so müssen Sie den Namen des Verzeichnisses direkt oben in der `Path`-Zeile eingeben, da eine Auswahl über den Verzeichnisbaum nicht möglich ist. Wenn Sie direkt in der `Path`-Zeile ein Verzeichnis angeben, das noch nicht existiert, fragt das Installationsprogramm nach, ob es das Verzeichnis erstellen soll. Wenn Sie hier mit `No` antworten, kehrt das

♠ Auch wenn dies der Einschränkung auf Privatgebrauch widerspricht … Sie können hier selbstverständlich „privat“ eintragen

Installationsprogramm zum Auswahlfeld zurück, damit Sie ein anderes Verzeichnis auswählen können. Wenn Sie mit `Yes` antworten, kehrt das Installationsprogramm direkt zum ursprünglichen Installationsziel-Bildschirm zurück und erstellt das gewünschte Verzeichnis.

Wenn Sie bei der Auswahl das Wurzelverzeichnis eines beliebigen Laufwerks (also z. B. `C:\` oder `D:\`) ausgewählt haben (was im allgemeinen nicht sonderlich sinnvoll ist), dann hängt das Installationsprogramm an den von Ihnen gewählten Pfad noch das Unterverzeichnis `PGP` an, das Programm wird also in `C:\PGP` bzw. `D:\PGP` installiert. Leider zeigt Ihnen das Installationsprogramm diese Korrektur hier noch nicht an. Das nach der Auswahl auf dem Bildschirm angezeigte Installationsziel entspricht also nicht dem tatsächlich für die Installation benutzten Verzeichnis, wenn Sie das Wurzelverzeichnis eines Laufwerks als Installationsziel angegeben haben.

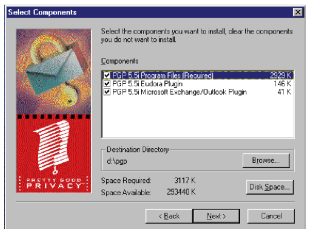


Abbildung 17.9: Auswahl der Komponenten (5.5.3i)

Mit `Next` bestätigen Sie die Auswahl des Zielverzeichnisses und gelangen zum Bildschirm für die Auswahl der zu installierenden Komponenten (Abb. 17.9). In diesem Bildschirm können Sie durch Anklicken der entsprechenden Kästchen mit den Häkchen bestimmte Komponenten von der Installation ausschließen, die Sie nicht benötigen.

Zur Auswahl stehen:

Die Programmdateien Diese werden immer benötigt; wenn diese Option nicht aktiviert ist, bringt das Installationsprogramm eine Fehlermeldung und fordert Sie auf, die Komponenten erneut auszuwählen. Auch wenn Sie nur eine einzelne Optionen, z. B. eine Erweiterung für ein E-Mailprogramm, zu einer bestehenden PGP-Installation nachinstallieren möchten, muß diese Option trotzdem angewählt bleiben.

Eudora Plugin Ein Zusatz für das E-Mailprogramm Qualcomm Eudora Light 3.05 oder neuer, der Ihnen die Funktionen von PGP in das E-Mailprogramm integriert. Wenn Sie Eudora als E-Mailprogramm benutzen, sollten Sie diese Option aktiviert lassen, sonst wählen Sie sie durch einen Mausklick auf das Häkchen ab.

Microsoft Exchange/Outlook Plugin Ein Zusatz, der Ihnen die Funktionen von PGP in das E-Mailprogramm Exchange oder Outlook integriert. Wenn Sie Outlook oder Exchange als E-Mailprogramm benutzen (nicht mit Outlook Express, das mit dem Microsoft Internet Explorer ausgeliefert wurde, verwechseln), sollten Sie diese Option aktiviert lassen, sonst wählen Sie sie durch einen Mausklick auf das Häkchen ab.

Für das E-Mailprogramm Outlook Express gibt es ein gesondertes Plugin, das nicht über das Installationsprogramm von PGP installiert wird. Sie finden das gesonderte Plugin für Outlook Express als `Plugin/Win32/Outlook Express (PGP 5.5.3i)/PGPOEPlugin.zip` auf der beiliegenden CD.

Als Grundeinstellung sind alle Komponenten für die Installation ausgewählt.

Auf diesem Bildschirm zeigt Ihnen das Installationsprogramm nochmals das (jetzt vollständige und korrekte, s. o.) Zielverzeichnis für die Installation an. Sie können es mit `Browse` nochmals ändern, wenn Sie es sich anders überlegt oder vorhin einen Fehler gemacht haben. Das Verfahren ist dasselbe wie oben beschrieben.

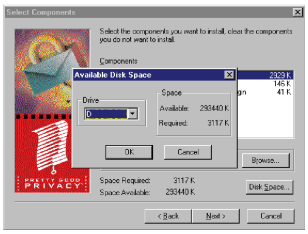


Abbildung 17.10: Anzeige des freien Platzes auf anderen Laufwerken (5.5.3i)

Außerdem zeigt Ihnen das Installationsprogramm an, wieviel Speicherplatz auf dem Laufwerk mit dem von Ihnen gewählten Zielverzeichnis benötigt wird und wieviel vorhanden ist. Über die Schaltfläche `Disk Space` kommen Sie in eine Maske (Abb. 17.10), wo Sie über ein Listenfeld diese Angaben auch für andere Laufwerke abrufen lassen können (wenn es andere Laufwerke in Ihrem System gibt). PGP installiert neben den Dateien im Zielverzeichnis auch noch einige Bibliotheken in den Systemverzeichnissen, also `C:\WINDOWS` und `C:\WINDOWS\SYSTEM` (unter Windows NT entsprechend; ganz exakt also `%SystemRoot%\` und `%SystemRoot%\System\`) sowie ggf. die Erweiterungen für die E-Mailprogramme in deren Programmverzeichnissen. Daraus folgt: Wenn sich das Zielverzeichnis der Installation z. B. auf Laufwerk `D:` befindet, wird auf `C:` trotzdem etwas Speicherplatz benötigt, vorausgesetzt, Windows ist auf `C:` installiert. Ob der Platz noch ausreicht, können Sie mit dieser Funktion überprüfen. Mit `OK` kommen Sie wieder zurück zum Komponentenauswahl-Bildschirm.

Abbildung 17.11: Zusammenfassung der Einstellungen (5.5.3i)

Durch Bestätigen mit `Next` gelangen Sie zum nächsten Bildschirm, wo Ihnen das Installationsprogramm die bisher gemachten Angaben zusammengefaßt anzeigt (Abb. 17.11). Hier haben Sie die Möglichkeit, die Eingabeprozedur durch Anklicken von `Back` nochmals zu durchlaufen und die Angaben gegebenenfalls zu korrigieren. Mit `Next` bestätigen Sie die gemachten Angaben und leiten den Kopiervorgang ein.

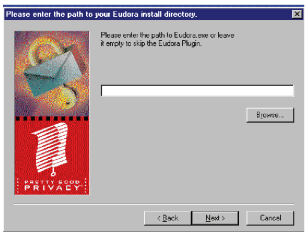


Abbildung 17.12: Mailprogramm nicht gefunden (5.5.3i)

Sollten Sie bei der Auswahl der Komponenten Erweiterungen für E-Mailprogramme ausgewählt haben, die das Installationsprogramm auf Ihrem Rechner nicht finden kann (z. B. weil sie nicht installiert sind oder weil sie sich auf einem Netzwerk-Laufwerk außerhalb des Rechners befinden, auf dem Sie die Installation durchführen), so fordert das Installationsprogramm Sie nun auf, den Pfad zu dem entsprechenden Programm anzugeben (Abb. 17.12). Sie können den Pfad direkt in das Feld eingeben, wenn er Ihnen bekannt ist, oder über die Schaltfläche `Browse` ein Fenster erhalten, in dem Sie das Verzeichnis auswählen können. Das Verfahren ist hierbei dasselbe wie bei der Auswahl des Installationsverzeichnisses weiter oben. Mit `OK` geht es zurück.

Abbildung 17.13: Installation läuft... (5.5.3i)

Wenn Sie das Eingabefeld leer lassen oder ein Verzeichnis auswählen, das das gesuchte Programm nicht enthält, wird die Erweiterung für das entsprechende E-Mailprogramm nicht installiert. Die eigentliche PGP-Installation funktioniert aber trotzdem, nur die Einbettung der PGP-Funktionen in das betreffende E-Mailprogramm findet nicht statt.

Nach der letzten Abfrage beginnt das Installationsprogramm mit dem Kopieren der Dateien. Dieser Vorgang kann eine Weile dauern, der Fortschritt wird Ihnen am Bildschirm angezeigt (Abb. 17.13). Nachdem der Kopiervorgang abgeschlossen ist, fragt

das Installationsprogramm nach, ob Sie bereits existierende Schlüsselbunde (von einer früheren PGP-Installation) benutzen möchten.

Wenn Sie diese Frage mit Nein beantworten, können Sie die nächsten fünf Absätze überspringen und gelangen direkt zum letzten Bildschirm.

Im allgemeinen ist es sinnvoll und einfacher, diese Frage mit Nein zu beantworten und die evtl. bereits vorhandenen Schlüssel zu einem späteren Zeitpunkt (nach Abschluß der Installation) zu importieren. Wenn Sie Schlüssel aus PGP-Versionen vor 5.x haben, in denen die Schlüsseldateien noch `pubring.pgp` und `secring.pgp` heißen, sollten Sie auf jeden Fall die Schlüssel später importieren, da ansonsten die Schlüsseldateien mit den Namen `pubring.pgp` bzw. `secring.pgp` weiterverwendet werden. Die Dateiendungen `.pgp` wurden von PGP bei Windows als „PGP-verschlüsselte Datei“ registriert, d. h. sie sind nicht als Schlüsselbund-Dateien erkennbar. Dies erhöht die Gefahr, daß die Dateien versehentlich gelöscht werden, außerdem reagiert Windows beispielsweise auf einen Doppelklick auf die Dateien falsch – es ruft nicht `PGPkeys` auf, sondern versucht, sie zu entschlüsseln. In den Versionen ab 5.x heißen die Dateien schon `pubring.pkr` und `secring.skr`, das obengenannte Problem tritt also nicht auf.

Wenn Sie die Frage mit Ja beantwortet haben, erhalten Sie ein Fenster, in dem Sie einige Grundeinstellungen von PGP ändern können, unter anderem auch die Pfade zu den Schlüsselbund-Dateien und der für die Zufallszahlen-Erzeugung benötigten Datei `randseed.bin`. Sie können die Pfade zu den entsprechenden Dateien entweder direkt in die Felder eingeben oder mit `Browse` zu einem Fenster gelangen, in dem Sie wie in einem „Datei öffnen“-Dialog den Pfad zu den entsprechenden Dateien ändern können (Abb. 17.14).

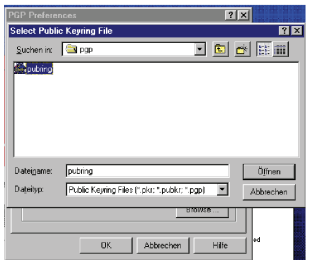


Abbildung 17.14: Auswahl zu importierender Schlüsselbunde (5.5.3i)

Zu beachten ist, daß das Installationsprogramm standardmäßig nach Dateien mit der Erweiterung `.pkr` (für „Public Key Ring“) sucht. Wenn Sie einen Schlüsselbund von PGP 2.6.x importieren wollen, der den Namen `pubring.pgp` trägt, so müssen Sie im Feld `Dateiname` die Angabe `*.pkr` durch `*` oder `*.pgp` ersetzen, sonst werden Ihre Dateien nicht zur Auswahl angezeigt. Mit der Schaltfläche `Öffnen` bestätigen Sie den Import des öffentlichen Schlüsselbundes.

Im Anschluß erscheint ein ähnliches Fenster für den privaten Schlüsselbund. Hier müssen Sie analog zu oben die Datei für den privaten Schlüssel auswählen und mit `Öffnen` bestätigen. Die Datei für den privaten Schlüsselbund in PGP 2.6.x heißt `secring.pgp`, hier muß also ebenfalls der Filtereintrag im Feld `Dateiname` geändert werden in `*.*` oder `*.pgp`.

Nach der Auswahl der Datei für den privaten Schlüsselbund erscheint eine Nachfrage, ob die Dateien in das PGP-Verzeichnis der aktuellen Installation kopiert werden sollen. Bitte beachten Sie hierzu die grundsätzlichen Überlegungen über die Sicherheit der Programme und der Schlüssel in Kapitel 7.4 auf Seite 65.

Wenn Sie sich entscheiden, die Schlüssel von der Festplatte zu verwenden, sollten Sie die Nachfrage mit `Ja` bestätigen. Ansonsten benutzt PGP die Schlüsselbunde vom ausgewählten Pfad (z. B. Diskette – aber bedenken Sie, daß Disketten eine sehr niedrige Lebenserwartung haben). Diese Einstellung und der Ort der Schlüsselbunddateien kann nachträglich verändert werden.

Nun haben Sie es geschafft. Als letzte Maske bringt Ihnen das Installationsprogramm die Mitteilung, daß die Installation abgeschlossen ist. Sie können nun durch Auswahl über das Anklicken der Häkchen festlegen, ob Sie das Programm `PGPkeys` für die Schlüsselverwaltung starten wollen und ob die Datei `readme.txt` mit Informationen über das Programm angezeigt werden soll.

Es empfiehlt sich, hier die Option `PGPkeys` abzuschalten. Bitte beachten Sie dazu den Abschnitt 17.1 auf Seite 146 am Anfang dieses Kapitels. Mit `Finish` beenden Sie die eigentliche Programminstallation.

17.4. Installation von PGP Freeware 6.0i für Windows

PGP Freeware 6.0i für Microsoft Windows 95/98/NT kommt in Form eines ausführbaren Programms `pgpfreeware60.exe`, das Sie auch auf der beiliegenden CD finden. Das Programm enthält die Installationsroutine und die benötigten Daten in komprimierter Form. Die Installation wird durch Ausführen des Programms, z. B. durch Doppelklick auf das Programmsymbol, gestartet.

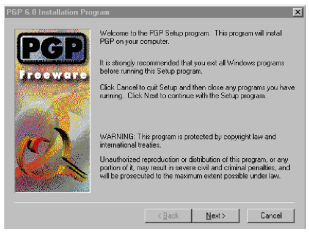


Abbildung 17.15: Der Begrüßungsschirm der Installation (6.0i)

Nachdem die Dekomprimierung abgeschlossen ist, erscheint ein Begrüßungsbildschirm (Abb. 17.15), den Sie mit einem Mausklick auf `Next` bestätigen.

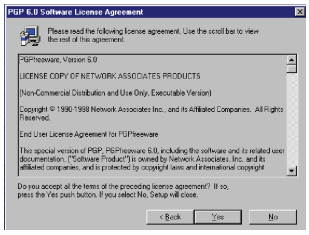


Abbildung 17.16: Der Lizenzvertrag (6.0i)

Als nächstes wird Ihnen der Lizenzvertrag angezeigt (Abb. 17.16), den Sie lesen und mit einem Mausklick auf `Yes` akzeptieren sollten. Wenn Sie ihn mit `No` ablehnen, beendet sich das Installationsprogramm (wie zu erwarten) ohne Installation von PGP.

Die Benutzung von PGP Freeware 6.0i ist nur für den nicht-kommerziellen Einsatz kostenlos, für den gewerblichen Einsatz darf die Freeware-Version nicht verwendet werden! Die Bedienung der gewerblichen Version ist im wesentlichen identisch zur hier beschriebenen Freeware-Version.

In Deutschland ist eine mögliche Bezugsquelle für kommerziell einsetzbare PGP-Versionen ART D'AMEUBLEMENT, Marktstraße 18, 33602 Bielefeld, 0521-6 55 66, Fax 0521-6 11 72, `www.pgpvertrieb.de`. Wenn Sie planen, PGP in Ihrer Firma einzusetzen, sollten Sie daran denken, eine Mitarbeiterschulung einzuplanen. Der Einsatz von Verschlüsselung ohne Hintergrundwissen kann wegen des trügerischen Sicherheitsgefühls gefährlicher sein als der Verzicht auf Verschlüsselung; deswegen betonen wir auch ständig, daß Sie den ersten Teil dieses Buches lesen sollten. Wir werden uns bemühen, alle uns bekannten seriösen und empfehlenswerten Schulungsangebote auf unsere Webseiten unter `http://www.foebud.org/` aufzunehmen.

Im nächsten Bildschirm geben Sie, falls Sie möchten, Ihren Namen und ggf. den Namen Ihrer Firma ein (Abb. 17.17). Die Eingabe eines Namens oder einer Firma ist nicht obligatorisch, die Installation funktioniert auch ohne diese Angaben. Das Installationsprogramm übernimmt als Voreinstellung die in Windows selbst angegebenen Benutzereinstellungen. Mit `Next` geht es weiter.

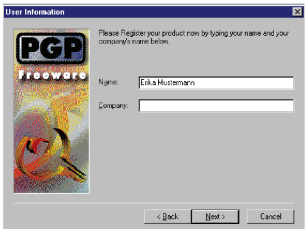


Abbildung 17.17: Die Benutzerdaten (6.0i)

Im folgenden Bildschirm werden Sie aufgefordert, ein Zielverzeichnis für die Installation anzugeben (Abb. 17.18). Sie können entweder durch Anklicken von `Next` die Voreinstellung von `C:\Program Files\Network Associates\PGP60` übernehmen oder durch Anklicken von `Browse` ein Fenster mit einem Verzeichnisbaum aufrufen (Abb. 17.19), wo Sie durch Auswählen aus dem Listenfeld `Drives` ein anderes Laufwerk als `C:` und durch Anklicken der Symbole für die Verzeichnisse ein anderes Verzeichnis auswählen können. Bestätigung und Rückkehr zum ursprünglichen Bildschirm mit `OK`.

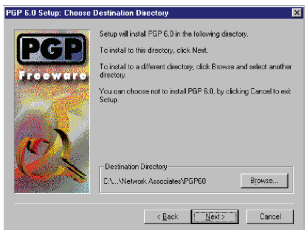


Abbildung 17.18: Auswahl des Installationsverzeichnisses (6.0i)

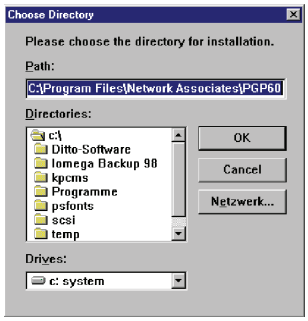


Abbildung 17.19: Verzeichniswahl im `Browse`-Fenster (6.0i)

Möchten Sie PGP in einem Verzeichnis installieren, das noch gar nicht existiert, so müssen Sie den Namen des Verzeichnisses direkt oben in der `Path`-Zeile eingeben. In diesem Fall fragt das Installationsprogramm nach, ob es das Verzeichnis erstellen soll. Wenn Sie hier mit `No` antworten, kehrt das Installationsprogramm zum Auswahlfeld zurück, damit Sie ein anderes Verzeichnis auswählen können. Wenn Sie mit `Yes` antworten, kehrt das Installationsprogramm direkt zum Installationsziel-Bildschirm zurück und erstellt das gewünschte Verzeichnis.

Das Installationsprogramm enthält bei der Verzeichnisauswahl einen Fehler, der etwas Verwirrung stiften kann. An den von Ihnen ausgewählten (oder direkt eingegebenen) Zielpfad

hängt das Installationsprogramm *immer* `PGP60` an. Leider wird Ihnen das nicht angezeigt. Sie können PGP also nur in ein Unterverzeichnis mit dem Namen `PGP60` installieren und lediglich bestimmen, wo sich dieses Verzeichnis befinden soll.

Angenommen, Sie möchten PGP im Verzeichnis `C:\PGP60` installieren, so dürfen Sie bei der oben beschriebenen Auswahl nur `C:\` als Installationsziel angeben. Wenn Sie `C:\PGP60` angeben, wird PGP im Verzeichnis `C:\PGP60\PGP60` installiert.

Das nach der Auswahl auf dem Bildschirm angezeigte Installationsziel entspricht also nicht dem tatsächlich für die Installation benutzten Verzeichnis.

Mit `Next` bestätigen Sie die Auswahl des Zielverzeichnisses und gelangen zum Bildschirm für die Komponentenauswahl (Abb. 17.20). In diesem Bildschirm können Sie durch Anklicken der entsprechenden Kästchen bestimmte Komponenten von der Installation ausschließen, die Sie nicht benötigen.

Zur Auswahl stehen:

Abbildung 17.20: Auswahl der Komponenten (6.0i)

Die Programmdateien Wenn diese Option nicht aktiviert ist, bringt das Installationsprogramm eine Fehlermeldung und fordert Sie auf, die Komponenten erneut auszuwählen. Auch wenn Sie einzelne Optionen auf eine bestehende Installation nachinstallieren möchten, muß diese Option angewählt bleiben.

Eudora Plugin Ein Zusatz für das E-Mailprogramm Qualcomm Eudora (Light) 3.05 und neuere Versionen, der Ihnen die Funktionen von PGP in das E-Mailprogramm integriert. Wenn Sie Eudora als E-Mailprogramm benutzen, sollten Sie diese Option aktiviert lassen, sonst sollten Sie die Option durch Klick auf das Häkchen abschalten.

Microsoft Exchange/Outlook Plugin Ein Zusatz für Microsofts E-Mailprogramm Outlook/Exchange, das PGP in das E-Mailprogramm integriert. Wenn Sie Outlook oder Exchange (nicht mit Outlook Express verwechseln) als E-Mailprogramm benutzen, sollten Sie diese Option aktiviert lassen, sonst wählen Sie die Option durch Klick auf das Häkchen ab.

Microsoft Outlook Express Ein Zusatz für Microsofts E-Mailprogramm Outlook Express, das die Funktionen von PGP integriert. Wenn Sie Outlook Express (nicht mit Outlook/Exchange verwechseln) als E-Mailprogramm benutzen, sollten Sie diese Option aktiviert lassen, sonst deaktivieren Sie die Option durch Klick auf das Häkchen.

Benutzerhandbuch als PDF-Datei (Portable Document Format) Sie benötigen eine Software, die PDF-Dateien darstellen kann, um das Benutzerhandbuch lesen zu können. Diese Software ist nicht im PGP-Paket enthalten. Ein Beispiel für eine Software, die PDF-Dateien anzeigen kann, ist die kostenlose Freeware Adobe Acrobat Reader, die auch auf der dem Buch beiliegenden CD enthalten ist.

Als Grundeinstellung sind alle Komponenten für die Installation ausgewählt.

Auf diesem Bildschirm zeigt Ihnen das Installationsprogramm nochmals das (diesmal vollständige und korrekte, s. o.) Zielverzeichnis für die Installation an, Sie können es mit `Browse` nochmals ändern, wenn Sie es sich anders überlegt oder vorhin einen Fehler gemacht haben.

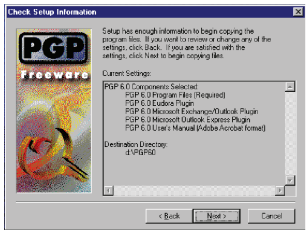


Abbildung 17.21: Zusammenfassung der Angaben (6.0i)

Außerdem zeigt Ihnen das Installationsprogramm an, wieviel Speicherplatz auf dem Laufwerk mit dem Zielverzeichnis benötigt wird und wieviel vorhanden ist. Über die Schaltfläche `Disk Space` kommen Sie in eine Maske, wo Sie über ein Listenfeld diese Angaben auch für andere Laufwerke abrufen können (wenn es andere Laufwerke in Ihrem System gibt). PGP installiert neben den Dateien im Zielverzeichnis auch noch einige Bibliotheken in den System-Verzeichnissen, auf den meisten Windows-Rechnern sind das `C:\WINDOWS` und `C:\WINDOWS\SYSTEM`. (Wie auf Seite 156 bemerkt, lauten die Verzeichnisnamen unter NT meistens anders.) Wenn sich das Zielverzeichnis der Installation auf Laufwerk `D:` befindet, wird auf `C:` also trotzdem etwas Speicherplatz benötigt. Ob der Platz noch ausreicht, können Sie mit dieser Funktion überprüfen.

Durch Bestätigen mit `Next` gelangen Sie zum nächsten Bildschirm (Abb. 17.21), wo Ihnen das Installationsprogramm nochmals die bisher gemachten Angaben zusammenfaßt und Ihnen die Möglichkeit ein-

räumt, die Eingabeprozedur durch Anklicken von `Back` nochmals zu durchlaufen und die Angaben zu korrigieren.

Sollten Sie bei der Auswahl der Komponenten Plugins für Programme ausgewählt haben, die das Installationsprogramm auf Ihrem Rechner nicht finden kann (z. B. weil sie nicht installiert sind oder weil sie sich auf einem Netzwerk-Laufwerk außerhalb Ihres Rechners befinden), so fordert das Installationsprogramm Sie nun auf, den Pfad zu den entsprechenden Programmen über einen Verzeichnisbaum auszuwählen. Wenn Sie hier `Abbrechen` wählen oder ein Verzeichnis auswählen, daß das gesuchte Programm nicht enthält, so erscheint eine Meldung, daß das Installationsprogramm das E-Mailprogramm nicht finden konnte und das entsprechende Plugin nicht installiert wird. Diese Meldung können Sie mit `OK` bestätigen.

Die eigentliche PGP-Installation funktioniert trotzdem, nur die Einbettung der PGP-Funktionen in das jeweilige E-Mailprogramm kann dann nicht stattfinden.

Nach der letzten Abfrage beginnt das Installationsprogramm mit dem Kopieren der Dateien. Dieser Vorgang kann eine Weile dauern. PGP zeigt Ihnen den Fortschritt an.

Wenn der Kopiervorgang abgeschlossen ist, fragt das Installationsprogramm nach, ob Sie bereits existierende Schlüsselbunde benutzen möchten. Wenn Sie bereits aus einer älteren PGP-Installation über einen Schlüssel bzw. über die öffentlichen Schlüssel anderer Nutzer verfügen, können Sie hier durch Anklicken von `Yes` den Import der Schlüsselbunde und das Kopieren in das PGP-Installationsverzeichnis einleiten. Sie können die vorhandenen Schlüssel aber auch später importieren, wenn Sie dies nicht jetzt tun möchten. Wenn Sie `No` anklicken, gelangen Sie direkt zum letzten Bildschirm und können die nächsten drei Absätze überspringen.

Wenn Sie `Yes` angeklickt haben, werden Sie zuerst aufgefordert, über das übliche Listenfeld den Pfad zu Ihrem öffentlichen Schlüsselbund anzugeben. Zu beachten ist, daß das Installationsprogramm standardmäßig nach Dateien mit der Erweiterung `.pkr` (für „Public Key Ring“) sucht. Wenn Sie einen Schlüsselbund von PGP 2.6.x importieren wollen, der den Namen `pubring.pgp` trägt, so müssen Sie im Feld Dateiname die Angabe `*.pkr` durch `*.*` oder `*.pgp` ersetzen, sonst werden Ihre Schlüsselbund-Dateien nicht zur Auswahl angezeigt. Mit `Öffnen` bestätigen Sie den Import des öffentlichen Schlüsselbundes.

Im Anschluß erscheint ein ähnliches Fenster für den privaten Schlüsselbund. Hier können Sie analog zum eben beschriebenen öffentlichen Schlüsselbund die Datei für den privaten Schlüssel `secring.pgp` auswählen und mit `Öffnen` bestätigen.

Nach der Auswahl der Datei für den privaten Schlüsselbund erscheint eine Nachfrage, ob die Dateien in das PGP-Verzeichnis der aktuellen Installation kopiert werden sollen. Bitte beachten Sie hierzu die grundsätzlichen Überlegungen über die Sicherheit der Programme und der Schlüssel in Kapitel 5.8 auf Seite 43. Wenn Sie sich entscheiden, die Schlüssel von der Festplatte zu verwenden, sollten Sie die Nachfrage mit `Yes` bestätigen. Ansonsten benutzt PGP die Schlüsselbunde vom ausgewählten Pfad (z. B. Diskette – was wir wegen der kurzen Lebensdauer von Disketten aber kaum empfehlen können; ZIP-Medien o. ä. sind eher anzuraten). Diese Einstellung kann nachträglich verändert werden.

Abbildung 17.22: Installation abgeschlossen – `PGPkeys` starten? (6.0i)

Nun haben Sie es geschafft. Als letzte Maske bringt Ihnen das Installationsprogramm die Mitteilung, daß die Installation abgeschlossen ist (Abb. 17.22). Sie können nun durch Auswahl der entsprechenden Häkchen festlegen, ob Sie das Programm `PGPkeys` für die Schlüsselverwaltung starten wollen und ob die Datei `readme.txt` mit Informationen über das Programm angezeigt werden soll. Bitte lesen Sie hierzu den Abschnitt 17.1 auf Seite 146. Mit `Finish` beenden Sie die eigentliche Programminstallation.

18. Schlüsselverwaltung – PGPkeys

PGPkeys ist der Teil von PGP, mit dem Sie die Verwaltung Ihrer öffentlichen und privaten Schlüssel vornehmen können. Hier können Sie neue Schlüssel erzeugen, die Vertrauenseinstellungen von Schlüsseln bearbeiten oder auch neue öffentliche Schlüssel Ihrem Schlüsselbund hinzufügen, damit Sie an deren Besitzer verschlüsselte Daten schicken können. Oder Sie können Schlüssel (eigene und fremde) exportieren, um Sie an andere Personen weiterzugeben.

Um PGP überhaupt sinnvoll benutzen zu können, benötigen Sie einen eigenen privaten Schlüssel. Deshalb stellen wir die Anleitungen zur Erstellung eines neuen Schlüssels an den Anfang dieses Kapitels, daran anschließend folgt in Abschnitt 18.2 auf Seite 173 die Anleitung zum Importieren bereits vorhandener Schlüsselbunde.

18.1. Schlüsselerzeugung

Um überhaupt sinnvoll mit PGP arbeiten zu können, benötigen Sie einen eigenen Schlüssel, mit dessen öffentlichen Teil Ihre Kommunikationspartner Daten an Sie verschlüsseln können und mit dessen privaten Teil Sie Daten signieren können. Wenn Sie noch nicht über einen PGP-Schlüssel aus einer früheren Installation verfügen, ist die Erzeugung eines neuen Schlüssels einer der ersten Schritte, die Sie unternehmen müssen, um PGP verwenden zu können.

Bevor Sie mit der Schlüsselerzeugung starten, möchten wir Sie jedoch auf den Abschnitt 17.1 auf Seite 146 hinweisen. Die Standardeinstellungen von PGP für Windows beinhalten eine Option Faster Key Generation, die unserer Meinung nach ein potentielles Sicherheitsrisiko bedeutet und daher vor der Schlüsselerzeugung abgeschaltet werden sollte. Wie Sie das tun können, können Sie in Abschnitt 17.1 nachlesen.

Wenn Sie die Voreinstellungen entsprechend geändert haben oder das nicht tun möchten, können Sie nun PGPkeys, das Programm zur Schlüsselverwaltung, entweder über den Eintrag im Windows-Startme-

nü, über das Menü von PGPtray oder über die entsprechende Funktion in PGPtools (erst ab Version 5.5.3i) aufrufen.

Wenn mit PGPkeys noch kein Schlüssel erzeugt worden ist und auch kein Schlüsselbund bereits bei der Installation importiert wurde, startet PGPkeys beim Aufruf automatisch das Programm Key Generation Wizard für die Erzeugung eines neuen Schlüssels.

Sie können aber auch, wenn die Schlüsselerzeugung nicht automatisch gestartet wird, aus dem laufenden PGPkeys-Programm heraus über den Menüpunkt Keys/New Key die Erzeugung eines neuen Schlüssels starten.

Der Ablauf ist für alle besprochenen Versionen von PGP für Windows sehr ähnlich, lediglich die graphische Gestaltung der Fenster weicht etwas ab, daher haben wir den Punkt Schlüsselerzeugung für alle Versionen (5.0i, 5.5.3i und 6.0i) zusammengefaßt.

Abbildung 18.1: Eingabe der persönlichen Daten für den neuen Schlüssel (5.5.3i)

Als erstes erhalten Sie ein Fenster mit einer kurzen Erläuterung, das Sie mit Weiter bestätigen, wenn Sie ein neues Schlüsselpaar erzeugen möchten. Wenn Sie PGPkeys beispielsweise aufgerufen haben, um einen bestehenden Schlüsselbund zu importieren, dann können Sie hier auf Abbrechen klicken. Wenn Sie Weiter gewählt haben, erhalten Sie ein Fenster, in dem Sie Ihren Vor- und Nachnamen sowie Ihre E-Mail-Adresse angeben sollen (Abb. 18.1). Der Name kann natürlich auch die Bezeichnung einer Organisation sein, wenn der Schlüssel für diese Organisation statt für Sie persönlich dienen soll. Die E-Mail-Adresse sollten Sie auf jeden Fall korrekt angeben, das erleichtert die Bedienung für Ihre Kommunikationspartnerinnen sehr, da PGP dann die Verschlüsselung automatisch vornehmen kann, wenn es einen zur E-Mail-Adresse passenden Schlüssel findet. Sollten Sie den Schlüssel für mehrere E-Mail-Adressen benutzen wollen, so geben Sie hier einfach die Adresse an, die am häufigsten genutzt wird. Sie können später weitere E-Mail-Adressen hinzufügen und auch nicht mehr benutzte wieder löschen, ohne einen neuen Schlüssel erzeugen zu müssen.△ Mit Weiter bestätigen Sie die Eingabe und gelangen zum nächsten Fenster.

△ Bitte beachten Sie, daß hierbei keine Adressen bei anderen Leuten gelöscht werden, die Ihren Schlüssel bereits auf dem Rechner haben.

Im folgenden Fenster müssen Sie den gewünschten Algorithmus (den „Typ“) des zu erzeugenden Schlüssels angeben (DSS/ElGamal⊖ oder RSA).

Falls Sie PGP 6.0i benutzen, betrifft Sie dieser Abschnitt nicht. Diese Version kann nur mit DSS/ElGamal-Schlüsseln arbeiten, daher entfällt die Auswahl.

Wenn Sie mit PGP 5.0i arbeiten, erscheint das Fenster zwar, Sie können allerdings die Option RSA nicht anwählen, da PGP 5.0i die Erzeugung von RSA-Schlüsseln nicht unterstützt (mit importierten RSA-Schlüsseln arbeiten kann es aber).

Wenn Sie PGP 5.5.3i benutzen und die Auswahl haben, dann lesen Sie bitte hierzu die allgemeinen Erläuterungen in Abschnitt B auf Seite 271 gegen Ende dieses Buches über die verschiedenen Schlüsselarten nach. Wenn die Möglichkeit gegeben ist, dann ist es sinnvoll, sowohl einen RSA- als auch einen DSS/ElGamal-Schlüssel zu erzeugen, denn nur so können Sie mit allen anderen Benutzerinnen von PGP kommunizieren (Benutzer älterer PGP-Versionen vor 5.x können nicht an DSS/ElGamal-Schlüssel verschlüsseln, Benutzer von PGP Version 6.0i können keine RSA-Schlüssel mehr verwenden).

Mit `Weiter` bestätigen Sie Ihre Auswahl.

Abbildung 18.2: Auswahl der Schlüssellänge (5.5.3i)

Im folgenden Fenster (jetzt wieder für alle Versionen) müssen Sie eine Länge für den zu erzeugenden Schlüssel angeben (Abb. 18.2). Prinzipbedingt sieht dieses Fenster bei der Erzeugung eines RSA-Schlüssels unter PGP 5.5.3i etwas anders aus, die Logik ist jedoch dieselbe. Grundsätzlich gilt: je länger der Schlüssel, um so sicherer, aber auch um so langsamer die Ver- und Entschlüsselungsvorgänge. Da die 32-Bit Windows-Versionen, die Systemvoraussetzung für PGP für Windows sind, aber sowieso nur auf Rechnern ab 386er Prozessoren aufwärts lauffähig sind, dürfte diese Verzögerung nicht allzu gravierend ausfallen. Sie sollten heutzutage keine Schlüssellängen von weniger als 1024 Bit mehr erzeugen, die Voreinstellung von 2048 Bit ist vielleicht ein wenig paranoid, dürfte aber normalerweise eine gute Wahl sein. Die gewünschte Schlüssellänge wählen Sie durch einen Mausklick in das runde Feld vor dem Text an. Mit `Weiter` bestätigen.

⊖ von PGP Inc. DSS/DH genannt, vgl. die Fußnote auf Seite 272

Im folgenden Fenster können Sie ein Ablaufdatum angeben, ab dem der Schlüssel nicht mehr gültig ist (Abb. 18.3).

Abbildung 18.3: Ablaufdatum des neuen Schlüssels (5.5.3i)

Für die meisten Anwendungen dürfte es sinnvoll sein, die Voreinstellung zu übernehmen, daß der Schlüssel unbegrenzt gültig ist. Wenn Sie einen Schlüssel erzeugen wollen, den Sie nur während einer bestimmten Zeitspanne benutzen möchten, dann geben Sie hier das „Haltbarkeitsdatum“ an. Näheres zur Gültigkeitsdauer von Schlüsseln finden Sie in Abschnitt 4.1 auf Seite 20.

Bei PGP 5.0i erfolgt die Angabe der Gültigkeit in Tagen ab dem Erzeugungsdatum des Schlüssels (max. 2000 Tage).

Bei PGP 5.5.3i können Sie ein Datum eingeben, an dem der Schlüssel ungültig wird, wobei das Datum in US-amerikanischer Schreibweise erwartet wird, also Monat/Tag/Jahr. 02/05/1999 ist also der 5. Februar, nicht der 2. Mai!

Bei PGP 6.0i können Sie ebenfalls ein Datum eingeben, hier wird für das Format Ihre Windows-Einstellung abgefragt; wenn Sie Windows beigebracht haben, daß Sie in Deutschland leben, können Sie also Tag/Monat/Jahr eingeben.

Wenn Sie ein Ablaufdatum für den Schlüssel angeben, dann können Sie oder andere Personen nach diesem Datum den Schlüssel nicht mehr für die Verschlüsselung von Daten verwenden. Auch das Signieren mit diesem Schlüssel ist Ihnen nach Ablauf des „Haltbarkeitsdatums“ nicht mehr möglich. Sie können jedoch Daten, die während der Gültigkeitsdauer mit diesem Schlüssel verschlüsselt wurden, auch nach Ablauf der Gültigkeitsdauer wieder entschlüsseln und andere Benutzerinnen können Signaturen, die mit diesem Schlüssel erzeugt wurden, auch nach Ablauf des Gültigkeitsdauer noch überprüfen.

Mit `Weiter` bestätigen Sie die gemachten Angaben und gelangen zum nächsten Fenster, in dem Sie aufgefordert werden, ein Mantra einzugeben (Abb. 18.4). Das Mantra ist der letzte Schutz Ihrer vertraulichen Daten, wenn Ihr privater Schlüssel jemandem in die Hände fällt. Daher sollte das Mantra nicht zu einfach zu erraten sein (der Name Ihres Hundes ist für jemanden, der Sie kennt, nicht schwer zu erraten, genauso wenig wie Ihre Telephonnummer oder Ihr Autokennzeichen). Es darf auch nicht zu kurz sein, 2 oder 3 Zeichen sind deutlich zu wenig. Ande-

rerseits sollte es nicht zu lang sein, da Sie es häufiger eingeben müssen, denn Sie benötigen es für jede Entschlüsselung und jedesmal, wenn Sie etwas signieren möchten.♣

Gut geeignet sind Mantras, die außer Buchstaben (große und kleine) auch Ziffern und Satzzeichen enthalten. Sie müssen das Mantra zweimal eingeben, um sicherzugehen, daß Sie sich nicht vertippt haben. Sie könnten sonst mit Ihrem neuen Schlüssel nichts anfangen, wenn Ihnen das Mantra aufgrund eines Tippfehlers bei der Eingabe nicht bekannt wäre. Näheres zur Wahl eines Mantras finden Sie in den Abschnitten 5.1 auf Seite 34 und 7.4 auf Seite 65. Beachten Sie bitte, daß Sie nicht vom ersten zum zweiten Fenster wechseln können, indem Sie RETURN tippen – die Tabulatortaste oder die Maus sind gefragt.

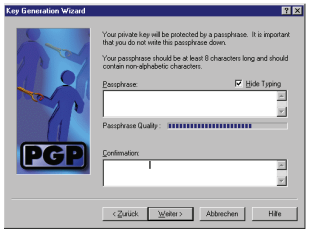


Abbildung 18.4: Eingabe des neuen Mantras (6.0i)

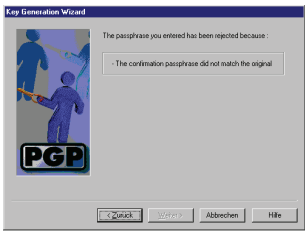


Abbildung 18.5: Kontrolleingabe des Mantras fehlgeschlagen (6.0i)

PGPkeys zeigt Ihnen in Form einer Fortschrittsanzeige die „Qualität“ des Mantras an. Diese Anzeige ist allerdings praktisch völlig willkürlich, Sie richtet sich nur nach Länge und Mischung von Zeichen, kann aber natürlich nicht Ihr persönliches Umfeld berücksichtigen. Da aber fünf zufällige Zeichen bestehend aus Groß- und Kleinbuchstaben, Ziffern und Satzzeichen ein besseres Mantra sind als z. B. der zwanzig Zeichen lange Name Ihrer Mutter, muß die Anzeige immer willkürlich bleiben. Selbst eine Eingabe von 32 a oder Ihrer eigenen E-Mail-Adresse genügt, um die Fortschrittsanzeige zufriedenzustellen. Mit Weiter bestätigen Sie das eingegebene Mantra.

Sollten die Eingaben im ersten und zweiten Feld nicht identisch gewesen sein, so zeigt Ihnen das Schlüsselerzeugungsprogramm eine entsprechende Fehlermeldung an (Abb. 18.5) und fordert Sie auf, das Mantra nochmals zweimal einzugeben. Erst wenn die beiden Eingaben identisch waren, gelangen Sie zum nächsten Fenster.

♣ PGP kann das Mantra auch zwischenspeichern – hierbei ist die Gefahr allerdings sehr groß, daß es im virtuellen Speicher auf der Festplatte landet, denn der Systemaufruf, der das verhindern soll, ist erstens nur bei Windows NT vorhanden und zweitens selbst dort wirkungslos.

Ist das von Ihnen eingegebene Mantra kürzer als acht Zeichen, so zeigt Ihnen PGP eine Meldung an, daß Ihr Mantra als nicht sicher bewertet wird und daß Sie ein längeres Mantra wählen sollten. Die Schlüsselerzeugung geht aber trotzdem normal weiter, wenn Sie mit `Weiter` bzw. mit `Accept bad passphrase` (bei PGP 5.0i) bestätigen. Mit `Zurück` bzw. mit `Cancel` kommen Sie zurück zum Mantra-Eingabefenster, wo Sie ein längeres Mantra eingeben können. Das Mantra kann später jederzeit abgeändert werden.

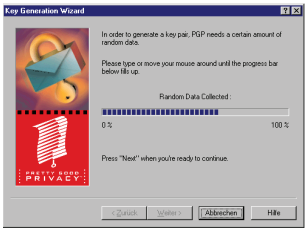


Abbildung 18.6: PGP sammelt Zufallsdaten (5.5.3i)

PGP benötigt für die Erstellung eines Schlüssels einige zufällige Zahlen. Diese übernimmt PGP zum Teil aus Ihren vorangegangenen Tastatureingaben (z. B. aus den Zeiten zwischen Tastenanschlägen, nicht einfach nur aus den eingegebenen Werten, denn dies ließe sich ja nachvollziehen). Wenn PGP in den vorangegangenen Masken nicht genügend Zufallsdaten gewinnen konnte, erscheint nun ein Fenster, in dem Sie aufgefordert werden, auf Ihrer Tastatur herumzutippen oder Ihre Maus zu bewegen, bis PGP genügend Zufallsdaten gesammelt hat (Abb. 18.6). Der Status wird dabei in einer Fortschrittsanzeige dargestellt, Sie können das Fenster erst dann mit `Weiter` beenden, wenn PGP genügend Daten gesammelt hat, vorher ist die Schaltfläche deaktiviert. Näheres zum Sammeln echter Zufallsdaten finden Sie in Abschnitt 4.3 auf Seite 21.

Abbildung 18.7: Schlüsselberechnung läuft . . . (5.5.3i)

Nun beginnt PGP mit der Berechnung der Parameter Ihres Schlüssels. Diese Berechnung kann recht lange dauern (wirklich lange), insbesondere auf nicht allzu schnellen Maschinen und ganz besonders dann, wenn Sie unseren Rat am Anfang des Abschnitts beherzigt haben, die Option der schnellen Schlüsselerzeugung der Sicherheit zuliebe abzuschalten. Also: nicht die Geduld verlieren! PGP zeigt Ihnen im Fenster an, was es gerade berechnet (Abb. 18.7). PGP zeigt in manchen Versionen zuerst die Berechnung der zweiten, dann die Berechnung der ersten Primzahl an . . . Bei PGP 5.0i und PGP 6.0i werden Sie während der Wartezeit mit einer kleinen animierten Graphik unterhalten. Wenn der Schlüssel fertig

berechnet ist, erscheint die entsprechende Meldung im Fenster und die Schaltfläche `Weiter` ist wieder verfügbar.

Im nächsten Fenster werden Sie gefragt, ob Sie Ihren neu erzeugten Schlüssel gleich an einen Key-Server im Internet schicken möchten (Abb. 18.8). Diese Option ist nur dann sinnvoll, wenn Sie über einen Internetzugang verfügen und dieser gerade aktiviert ist. Im allgemeinen wird das nicht der Fall sein (es sei denn, Sie sind an einer Standleitung angeschlossen), also werden Sie das Fenster normalerweise mit `Weiter` verlassen und die Option `Send my key to the default server now` (bei PGP 6.0i heißt die Option `Send my key to the root server now`) deaktiviert lassen. Wenn Sie es wünschen, können Sie das Verschicken später jederzeit nachholen.

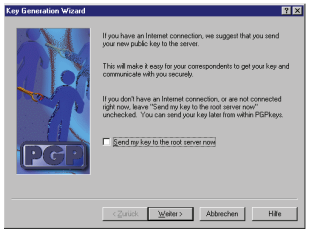


Abbildung 18.8: Schlüssel an Server senden? (6.0i)

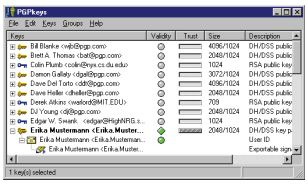


Abbildung 18.9: `PGPkeys` (5.5.3i)

Damit ist die Schlüsselerzeugung abgeschlossen, PGP beglückwünscht Sie noch zur erfolgreichen Erzeugung. Mit `Fertigstellen` beenden Sie das Programm zur Schlüsselerzeugung und das Fenster von `PGPkeys` erscheint auf dem Bildschirm (Abb. 18.9). Hier sehen Sie zum einen Ihren neu erzeugten Schlüssel, zum anderen, je nach Version, evtl. eine ganze Reihe von öffentlichen Schlüsseln anderer Personen. Es handelt sich hierbei um Schlüssel von Personen, die an der Entwicklung oder Verbreitung von PGP beteiligt sind.

Wenn Sie, wie oben angeregt, gleich zwei Schlüssel erzeugen möchten (DSS/ElGamal und RSA), so können Sie die gesamte Prozedur durch Aufruf von `Keys/New Key` aus dem Menü von `PGPkeys` nochmals durchlaufen.

Wenn Sie `PGPkeys` nach der Erzeugung eines neuen Schlüssels verlassen, fordert Sie PGP auf, den Schlüsselbund mit Ihrem neuen Schlüssel bzw. Ihren neuen Schlüsseln zu sichern. Diesen Rat sollten Sie beherzigen, denn wenn die Datei mit Ihrem Schlüsselbund z. B. durch versehentliches Löschen oder einen Defekt Ihres Datenträgers verloren geht, können Sie keine Daten mehr lesen, die an Sie verschlüsselt wurden. Sie können dann nicht einmal den verlorengegangenen Schlüssel zurückziehen, denn auch dafür benötigen Sie Ihren Schlüssel (bei PGP 6.0i kann

der sogenannte *designated revoker* den Schlüssel für Sie zurückziehen, wenn Sie einen solchen festgelegt haben). Wenn Sie der Aufforderungen durch Klicken auf `Save Backup Now` nachkommen, erscheint der normale Windows-Dialog `Speichern unter`, wo Sie für die beiden Dateien `secring.skr` (private(r) Schlüssel) und `pubring.pkr` (öffentlicher Schlüsselbund) einen Speicherort wählen können. Dieser sollte nach Möglichkeit *nicht* auf dem selben physikalischen Datenträger liegen wie Ihre Originale, sonst ist beispielsweise bei einem Festplattendefekt wenig gewonnen. Übrigens können Sie selbstverständlich auch die Dateien `secring.skr` und `pubring.pkr` direkt sichern, beispielsweise auf Diskette oder (falls Sie einen Brenner haben) CD. Die Sicherungskopie sollten Sie natürlich sorgfältig verwahren, nach Möglichkeit nicht direkt neben Ihrem Rechner – CDs überstehen Feuer nicht.

Damit haben Sie es geschafft, die Schlüsselerzeugung ist abgeschlossen.

18.2. Importieren und Hinzufügen von Schlüsseln zu Ihrem Schlüsselbund

Um PGP sinnvoll verwenden zu können, reicht es nicht aus, wenn Sie einen neuen Schlüssel für sich erzeugen. Um Daten an Ihre Kommunikationspartner verschlüsseln zu können, benötigen Sie deren öffentliche Schlüssel. Um die Signaturen, also die digitalen Unterschriften, anderer Leute unter Daten überprüfen zu können, benötigen Sie ebenfalls die öffentlichen Schlüssel der Unterzeichnenden.

Um diese Schlüssel verwenden zu können, müssen Sie sie in Ihren Schlüsselbund mit öffentlichen Schlüsseln aufnehmen.

Aber auch, wenn Sie bereits über einen Schlüsselbund mit privaten oder öffentlichen Schlüsseln aus einer vorherigen PGP-Installation verfügen, den Sie weiterhin verwenden möchten, ist es sinnvoll, die darin enthaltenen Schlüssel in PGP zu importieren. Schließlich möchten Ihnen nicht jedesmal alle Ihre Kommunikationspartnerinnen einen neuen Schlüssel zukommen lassen, wenn Sie PGP neu installieren.

Eine Einschränkung gibt es beim Import von Schlüsseln in PGP für Windows aber: Wenn der Schlüssel, den Sie importieren möchten, ein RSA-Schlüssel ist, so können Sie ihn in PGP Freeware 6.0i zwar importieren, dort aber nicht benutzen, da die Freeware-Version von PGP 6.0 RSA-Verschlüsselung nicht unterstützt (PGP 6.0i zeigt Ihnen das an, in-

dem es die Daten der RSA-Schlüssel in kursiver Schrift darstellt, siehe Abb. 18.10). In den anderen Versionen von PGP für Windows können Sie RSA-Schlüssel nach dem Import in PGP für Windows weiter verwenden.

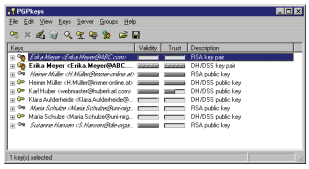


Abbildung 18.10: PGPkeys: RSA-Schlüssel kursiv (6.0i)

Um einen Schlüssel (oder einen ganzen Schlüsselbund) importieren zu können, benötigen Sie zuerst die Daten, die den Schlüssel ausmachen. Diese können z. B. in Form einer Datei auf einer Diskette vorliegen, die Sie von Ihren Kommunikationspartnerinnen bekommen haben, oder Sie haben den neuen Schlüssel von Ihren Kommunikationspartnern als E-Mail zugeschickt bekommen.

Grundsätzlich gibt es zwei Möglichkeiten, wie Sie solche Schlüssel in PGP importieren können: Über die entsprechenden Funktionen von PGPkeys oder über die Windows-Zwischenablage („Clipboard“).

18.2.1. Allgemeines zum Schlüsselimport und wenn Sie die zu importierenden Schlüssel als eigene Datei vorliegen haben

Wenn der Schlüssel, den Sie importieren möchten, als Datei vorliegt, gestaltet sich der Import sehr einfach. Wenn die Datei eine Endung hat, die PGP für Windows registriert hat (*.asc, *.pgp, *.pkr und *.skr), dann müssen Sie lediglich im Windows-Explorer auf das Symbol der jeweiligen Datei doppelklicken. Oder Sie klicken mit der rechten Maustaste auf das Symbol der Datei und wählen aus dem erscheinenden Kontextmenü den Punkt PGP/Decrypt/Verify (bei Dateien mit der Endung .asc oder .pgp) bzw. PGP/Add Keys to Keyring (bei Dateien mit der Endung .pkr oder .skr). PGP erkennt am Dateityp automatisch, daß die Datei Schlüssel enthält und startet den Import.

Dasselbe gilt, wenn Sie PGP-Schlüssel als an E-Mails angefügte Dateien, sog. Attachments, erhalten. Hier genügt es, im Anzeigefenster Ihres E-Mailprogramms mit einem Doppelklick auf das Symbol der entsprechenden angehängten Datei den Schlüsselimport zu starten.

Wenn die Datei keine der oben genannten Endungen trägt, dann ist es einfacher, zuerst PGPkeys zu starten und dann den Import in PGPkeys über den Menüpunkt Keys/Import vorzunehmen. Nach Auswahl des Menüpunktes müssen Sie über ein Datei öffnen-Fenster von Windows die Datei auswählen, die die zu importierenden Schlüssel enthält.

Abbildung 18.11: Private Schlüsel importiert – Vertrauenseinstellungen prüfen! (5.5.3i)

Wenn Sie private Schlüssel importieren möchten (weil Sie z. B. von PGP 2.6.xi auf PGP für Windows umsteigen möchten und Ihren RSA-Schlüssel behalten wollen), dann sollten Sie in jedem Fall den Weg über das Menü `Keys/Import` von `PGPkeys` wählen. Der Import von privaten Schlüsseln über die Explorer-Funktion arbeitet unzuverlässig. Manchmal funktioniert der Import von privaten Schlüsseln auf diesem Weg, aber unter Umständen wird dabei nur der *öffentliche* Teil des Schlüssels importiert. Damit können Sie die an Sie verschlüsselten Nachrichten nicht lesen! Wenn Sie die Datei mit dem Schlüssel über das Menü von `PGPkeys` importieren und die Datei einen oder mehrere private Schlüssel enthält, dann wird Ihnen eine Meldung angezeigt, daß die ausgewählten Daten einen oder mehrere private Schlüssel enthalten, für die Sie die Gültigkeits- und Vertrauenseinstellungen manuell setzen müssen (Abb. 18.11, nicht bei PGP 5.0i). Das liegt daran, daß ein privater Schlüssel ja niemand gegeben werden soll, er kann also normalerweise auch von niemand anderem unterschrieben sein und ist damit beim Import in das Programm erst einmal ungültig. Aber Sie können ja beurteilen, ob es Ihr eigener Schlüssel ist oder nicht und sollten, wenn dies der Fall ist, dann nach dem Import die Einstellungen auf *implicit trust* ändern (s. u.). Wenn Sie Schlüssel über einen anderen Weg als das Menü von `PGPkeys` importieren, erscheint diese Meldung nicht, auch wenn die Daten private Schlüssel enthalten. PGP rechnet wohl (hoffentlich zu Recht) nicht damit, daß private Schlüssel per E-Mail verschickt werden. Die Vertrauenseinstellungen sollten Sie trotzdem kontrollieren.

Der jetzt folgende Ablauf ist für alle in diesem und den nachfolgenden Absätzen beschriebenen Varianten des Schlüsselimports gleich, und gilt daher auch für die unten beschriebenen Verfahren:

PGP 5.0i importiert alle in den jeweiligen Daten enthaltenen Schlüssel automatisch ohne Rückfrage und zeigt Ihnen nur eine Meldung an, daß es die Schlüssel erfolgreich importiert hat. Dies gilt sowohl für öffentliche wie für private Schlüssel.

PGP 5.5.3i und PGP 6.0i zeigen Ihnen zuerst ein Fenster an, in dem die in den Dateien enthaltenen Schlüssel aufgelistet werden (Abb. 18.12). Sie können dann durch Markierung mit der Maus in der Liste diejenigen Schlüssel aussuchen, die Sie in Ihren Schlüsselbund importieren

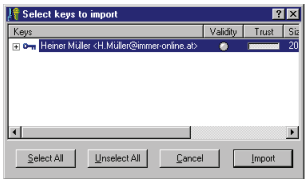


Abbildung 18.12: Auswahl zu importierender Schlüssel (5.5.3i)

möchten. Wenn Sie alle erhaltenen Schlüssel importieren möchten, können Sie auf die Schaltfläche `Select all` klicken. Wenn Sie nur ein oder zwei Schlüssel aus einer längeren Liste importieren möchten, können Sie mit der Schaltfläche `Unselect all` zuerst alle abwählen, um dann die gewünschten Schlüssel zu markieren. Wenn Sie alle gewünschten Schlüssel markiert haben, klicken Sie auf die Schaltfläche `Import`, um die Schlüssel in Ihren Schlüsselbund zu übernehmen. Wenn Sie auf die Schaltfläche `Cancel` klicken, wird der Schlüsselimport abgebrochen, ohne daß Schlüssel importiert werden.

18.2.2. Hinzufügen von Schlüsseln über die Windows-Zwischenablage Hinzufügen von Schlüsseln aus E-Mails, wenn Ihr E-Mailprogramm nicht von PGP unterstützt wird

Wenn Sie den Schlüssel im Text einer E-Mail erhalten haben – nicht als angefügte Datei (Attachment), hierzu lesen Sie bitte Abschnitt 18.2.1 auf Seite 174 – und die PGP-Funktionen nicht in Ihrem E-Mailprogramm integriert sind, ist der Schlüsselimport über die Windows-Zwischenablage der einfachste Weg. Hierzu markieren Sie in Ihrem E-Mailprogramm mit der Maus oder der Tastatur einfach in dem betreffenden Text den Abschnitt, der den PGP-Schlüssel enthält. Der Beginn dieses Blocks wird durch die Zeile

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

angezeigt, gefolgt von einer Zeile mit der Angabe der PGP-Version, aus der der betreffende Schlüssel exportiert wurde. Nach einem längeren Block von scheinbar sinnlosen Zeichen endet der PGP-Schlüssel mit der Zeile

```
-----END PGP PUBLIC KEY BLOCK-----
```

Ein solcher Schlüsselblock sieht so aus (gekürzt):

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGPfreeware 5.5.3i for non-commercial use <http://www.pgpi.com>
Comment: Andreas-Kommentar

mQGiBDejlngRBADaGqkSf88RMUm3FKpyX5sujHaHThjFR2F1aonWsFLxyq9Xm6Rd
aOO4a4DOIIpl9gyc4M8d9kj1a3WB+FUXgbhk6DZ1ZURJDZAoVBLP3La76jSxhmae
PO2lrw6JFZOWDHU4AnS3CiU6lI/Uv6p3uKmlcp/ONSk72UZrPhGbEZy+mQCg/yei
WB38YR7P5+pA527dce1RCbMEAI4ChMz4D3I5eJTo2OPOblZPkbiLd2F/SMdROYY
t3GLN4WALnVRFPQ78Xv62xXSkutDNfXtc9uSL5nSBPKIave2rih805ntA7x+4JS9
...
cYKlT1mcFCPWW8rGPnW/ARnyD94cGHZyyBp6tMdMq/HeUoAdhzMI6gnhKeEypxPj
dZOdB9akDOqm10ceNJ4ATaNbOsG4kJ9yQIv9qLcjXhv4XRh7dEeJULb6YwxHXloa
YiCGMPVZEnn49yBvwG6thIkAPwMFGDejlnguxCV/QNoogBECVZkAoK5gyIAYQn6V
AgYuUvaXUM78Pg9gAJ9LAwQscCog7u21v9ks/OgwKOL1Dw==
=TPXs
-----END PGP PUBLIC KEY BLOCK-----
```

Die hier beschriebenen Anfangs- und Endzeilen müssen Sie mit markieren, sonst erkennt PGP nicht, daß es sich bei den Daten um einen PGP-Schlüssel handelt.

Kopieren Sie mit den üblichen Windows-Funktionen (z. B. rechte Maustaste, im dann erscheinenden Menü den Punkt `Kopieren` auswählen oder `Strg-C` für Kopieren auf der Tastatur drücken) den markierten Text in die Zwischenablage. Wenn Sie mehr als nur den PGP-Schlüssel, also zum Beispiel den gesamten Text der E-Mail in die Zwischenablage kopiert haben, stört sich PGP daran nicht, es findet die Schlüsseldaten, die für PGP wichtig sind, automatisch aus dem übrigen Text heraus. Sie können also auch der Einfachkeit halber den gesamten Text der E-Mail markieren.

Wenn Sie den Text, der den PGP-Schlüssel enthält, in die Zwischenablage kopiert haben, können Sie den Schlüsselimport über den Menüpunkt `Add Key from Clipboard` im Menü von `PGPtray` anstoßen. PGP durchsucht nun automatisch die Daten in der Zwischenablage nach PGP-Schlüsseln.

Danach erscheint das Fenster mit den in der Datei gefundenen Schlüsseln zur Auswahl und es geht weiter wie in Abschnitt 18.2.1 auf Seite 174 beschrieben.

18.2.3. Hinzufügen von Schlüsseln aus E-Mails wenn Ihr E-Mailprogramm von PGP unterstützt wird

Wenn Sie den Schlüssel im Text einer E-Mail erhalten haben (nicht als angefügte Datei, sog. Attachment, hierzu lesen Sie bitte Abschnitt 18.2.1 auf Seite 174) und Sie ein E-Mailprogramm benutzen, in dem die PGP-Funktionen per Plugin integriert sind, müssen Sie im Anzeigefenster der Nachricht, die einen zu importierenden Schlüssel enthält, lediglich auf die Schaltfläche `Add Key to Keyring` klicken oder aus dem Menü den Punkt `PGP/Add Key` anwählen, der Schlüssel wird dann automatisch importiert, es geht weiter wie in Abschnitt 18.2.1 beschrieben.

Wenn Sie PGP 5.5.3i oder PGP 6.0i installiert haben und die Option `Automatically Decrypt/Verify when opening messages` (Nachrichten beim Öffnen automatisch entschlüsseln und prüfen) aktiviert ist, dann startet PGP den Import von Schlüsseln, die in einer unverschlüsselten Datei enthalten sind, bereits beim Öffnen der E-Mail automatisch. Wenn sich der Schlüssel im Text einer verschlüsselten und/oder signierten Nachricht befindet, müssen Sie die Aufnahme in den Schlüsselbund allerdings doch wieder wie oben beschrieben aufrufen.

Der Grund dafür ist: Die Automatikfunktion prüft *einmal* beim Öffnen der Nachricht, ob darin für PGP relevante Informationen sind. Sie startet dann die entsprechende PGP-Funktion und beendet sich nach Abschluß dieser Funktion. Wenn sie beim Durchsuchen einen Schlüssel findet, startet sie den Schlüsselimport. Wenn die Nachricht verschlüsselt ist, startet die Automatikfunktion die Entschlüsselung und beendet sich. Das Ergebnis der Automatikfunktion wird *nicht nochmals* überprüft, daher wird ein Schlüssel in einer verschlüsselten Nachricht von der Automatikfunktion *nicht* bemerkt.

18.3. Exportieren von Schlüsseln

Gelegentlich wollen Sie Schlüssel aus `PGPkeys` wieder exportieren, um Sie an andere Leute weiterzugeben. Dies betrifft sowohl Ihren eigenen Schlüssel als auch andere Schlüssel in Ihrem Schlüsselbund. Wenn Sie einen Schlüssel als Datei abspeichern möchten, können Sie dies über den Menüpunkt `Keys/Export` erreichen, nachdem Sie den oder die Schlüssel in der Liste markiert haben, die Sie exportieren wollen (Sie können damit auch mehrere verschiedene Schlüssel in eine Datei expor-

tieren). Dasselbe erreichen Sie durch Klicken mit der rechten Maustaste auf den oder die markierten Schlüssel in der Liste und Auswählen des Menüpunktes `Export` aus dem erscheinenden Kontextmenü.

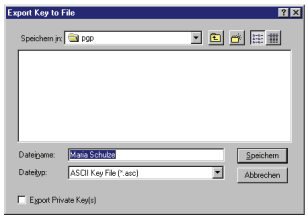


Abbildung 18.13: Auswahl der Zieldatei beim Schlüsselexport (5.5.3i)

In dem darauf erscheinenden Fenster wählen Sie ein Laufwerk, Verzeichnis sowie einen Dateinamen für die Datei aus (Abb. 18.13).

Bei PGP 5.5.3i und PGP 6.0i können Sie, wenn Sie für den Export einen privaten Schlüssel markiert hatten, über eine Option noch angeben, ob Sie den privaten Teil Ihres Schlüssels mit exportieren möchten. Standardmäßig ist diese Option immer abgewählt, es wird also nur der öffentliche Teil des Schlüssels exportiert, der bedenkenlos weitergegeben werden kann. Wenn Sie auch den privaten Teil exportieren möchten, aktivieren Sie die Option durch Klick auf das Kästchen. Sie sollten den privaten Teil Ihres Schlüssels *niemals* an andere Personen weitergeben! Wenn Sie einen Schlüssel für den Export ausgewählt haben, von dem sich nur der öffentliche Teil in Ihrem Schlüsselbund befindet (also ein Schlüssel von jemand anderem), ist diese Option (natürlich) nicht verfügbar.

Eine andere Exportmöglichkeit, wenn Sie Ihren Schlüssel z. B. in den Text einer E-Mail einfügen möchten, ist, den zu exportierenden Schlüssel mit der rechten Maustaste anzuklicken und den Menüpunkt `Copy` im erscheinenden Kontextmenü anzuklicken, hiermit wird der Schlüssel nicht in eine Datei, sondern in die Windows-Zwischenablage kopiert.

Danach wechseln Sie in das Anwendungsprogramm und an die Stelle, wo Sie den Schlüssel-Block einfügen möchten. Dort drücken Sie wieder die rechte Maustaste und wählen `Einfügen` bzw. `Paste` aus Ihrem Kontextmenü. Dann wird der Schlüssel als Textblock (wie auf Seite 177 gezeigt) eingefügt. Bei dieser Methode wird immer nur der öffentliche Teil von Schlüsseln kopiert, Sie können so keinen privaten Schlüssel kopieren.

18.4. Unterschreiben eines Schlüssels

Mit Ihrer Unterschrift unter einen Schlüssel bestätigen Sie, daß der Schlüssel tatsächlich zu der Person gehört, die in den Benutzernamen

des Schlüssels angegeben ist. Bitte lesen Sie hierzu die allgemeinen Erläuterungen im Abschnitt 7.1.2 auf Seite 60.

Wenn Sie sicher sind, daß der fragliche Schlüssel der angegebenen Person gehört, können Sie ihn unterschreiben. Dazu markieren Sie den Schlüssel, den Sie unterschreiben möchten, in der Liste mit einem Mausklick. Sie können dann entweder aus dem Menü von PGPkeys den Punkt Keys/Sign aufrufen oder mit der rechten Maustaste über das Kontextmenü den Punkt Sign auswählen. Mit Ihrer Unterschrift unter einen Schlüssel wird dieser Schlüssel für Sie automatisch gültig, das heißt er ist ausreichend beglaubigt.

Bei PGP 5.0i erscheint eine Meldung, in der noch einmal kurz erläutert wird, was das Unterschreiben eines Schlüssels bedeutet und die die Nachfrage stellt, ob Sie den Schlüssel wirklich unterschreiben möchten (Abb. 18.14). So etwas fehlt bei späteren Versionen leider. Außerdem können Sie, wenn Sie eine aktive Verbindung zum Internet haben, Ihre Signatur für den Schlüssel durch Auswahl der entsprechenden Option gleich an den eingestellten Keyserver schicken.

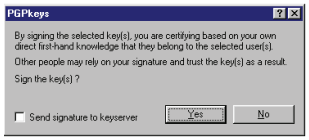


Abbildung 18.14: Erläuterungen zum Unterschreiben (5.0i)

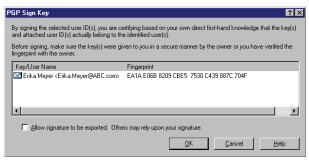


Abbildung 18.15: Schlüssel unterschreiben (5.5.3i)

PGP 5.5.3i und PGP 6.0i zeigen Ihnen in diesem Nachfragefenster den zu unterschreibenden Schlüssel mit dem zugehörigen Fingerabdruck (Fingerprint) an (Abb. 18.15). Aus irgendeinem Grund entfällt die Option Schicken zum Keyserver (bei PGP 6.0i ist es über die Grundeinstellungen möglich, dieses Verhalten anzuwählen; siehe Abschnitt 18.4 auf der vorherigen Seite). Dafür können Sie in diesen Versionen angeben, ob Ihre Unterschrift beim Exportieren des unterschriebenen Schlüssels mit exportiert werden soll oder nicht (die Standardeinstellung ist, daß die Unterschrift *nicht* wieder exportiert wird). Diese Option ist anscheinend dafür gedacht, daß Sie PGP mit einer nicht exportierbaren Unterschrift dazu bringen können, einen Schlüssel nicht zu monieren, der an sich nicht ausreichend bestätigt ist. Wir möchten *dringend* davon abraten, sie für diesen Zweck zu verwenden – die Warnung, die in so einem Fall erscheint, ist absolut berechtigt, gefälschte Schlüssel sind absolut keine Seltenheit; wenn Sie einen Schlüssel häufiger für private Kommunikation oder zum Überprü-

fen von Unterschriften verwenden, sollten Sie sich die Mühe machen, seine Authentizität zu überprüfen. In dem seltenen Fall, daß Sie beispielsweise den Fingerprint im Serviceteil der c't o. ä. abgedruckt finden, mag eine nicht exportierbare Signatur Sinn machen, aber Sie sollten nur Unterschriften unter Schlüssel setzen, wenn Sie auch bereit sind, diese weiterzugeben und dafür einzustehen. Ein weiteres Beispiel, wo nicht exportierbare Unterschriften Sinn machen, finden Sie vier Absätze weite im Text.

Bei PGP 6.0i haben Sie über die Schaltfläche `More Choices` noch weitere Einstellmöglichkeiten für die Signatur (Abb. 18.16).

Hier können Sie zum einen ein Verfallsdatum für die Unterschrift angeben, ab dem die Unterschrift nicht mehr gültig ist. Zum anderen haben Sie außer der normalen Unterschrift (exportierbar oder nicht exportierbar – wie oben beschrieben) noch die Möglichkeit, einem Schlüssel mit Ihrer Unterschrift die Eigenschaften *Meta-Introducer* oder *Trusted Introducer* zu verleihen.

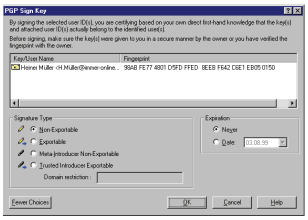


Abbildung 18.16: Weitere Optionen Schlüsselsignatur (6.0i)

Trusted Introducer (Vertrauenswürdiger Einführer) Diese Einstellung bedeutet, daß für Sie jeder Schlüssel gültig ist, der vom Vertrauenswürdigen Einführer mit dessen Schlüssel beglaubigt wurde. Von der Logik ist dies dasselbe, als wenn Sie einen Schlüssel mit der Vertrauenseinstellung „volles Vertrauen“ versehen. Der Unterschied besteht darin, daß die Angabe des Vertrauens, das Sie in die betreffende Person setzen, *mit exportiert* wird, Sie also allgemein bekanntgeben, daß die Person Ihrer Meinung nach absolut vertrauenswürdig ist. (Normalerweise werden Vertrauenseinstellungen nicht mit exportiert, es geht schließlich niemanden etwas an, wem Sie vertrauen und wem nicht.) Diese Einstellung ist hauptsächlich für Zertifizierungsstellen gedacht, die damit die Berechtigung zur Zertifizierung sozusagen delegieren können.

In PGP 5.5.3i können Sie diese Einstellungen zwar nicht selbst einem Schlüssel zuweisen, es erkennt Sie aber, wenn sie von einer anderen Version erzeugt wurden. Allerdings nennt PGP 5.5.3i den *Trusted Introducer* verwirrenderweise *Exportable Meta-Introducer*

(was nicht mit dem Meta-Introducer von PGP 6.0i verwechselt werden darf...).

Meta Introducer (Meta-Einführer) Ein Schlüssel mit dieser Einstellung ist ein vertrauenswürdiger Einführer, aber mehr: Diese Einstellung bedeutet, daß für Sie nicht nur jeder Schlüssel gültig ist, der vom Meta-Einführer mit dessen Schlüssel unterschrieben wurde, sondern darüber hinaus auch jeden Schlüssel als Vertrauenswürdigen Einführer betrachten, dem der Meta-Einführer dieses Attribut zuweist.

Diese Einstellung ist *nicht* exportierbar, beim Export wird daraus der „normale" Vertrauenswürdige Einführer.

Ein Beispiel: In Ihrem Betrieb haben alle Angestellten eigene PGP-Schlüssel, darüber hinaus gibt es Schlüssel für bestimmte Abteilungen. (Hierfür könnte die Funktionalität von PGP 6.0i interessant sein, mit der ein Schlüssel auf mehrere Rechner aufgeteilt werden kann). Die Schlüssel der Mitarbeiter sollen jeweils von einem Mitarbeiter der Personalabteilung unterschrieben werden. Eine technische Realisierung läßt sich so gestalten, daß jeder Mitarbeiter auf seinem Rechner den Schlüssel der Personalabteilung$^{\times}$ mit einer nicht exportierbaren Unterschrift$^{\diamondsuit}$ für gültig erklärt. Weiterhin wird dieser Schlüssel auf allen Firmenrechnern zum Meta-Einführer ernannt. Die Personalabteilung ihrerseits signiert die Schlüssel ihrer Mitarbeiter und markiert sie als vertrauenswürdige Einführer. Importiert nun ein Angestellter einen solchen Schlüssel, so wird er aufgrund der beiden Einstellungen (Abteilung: Meta-Einführer, Mitarbeiter: vom Abteilungsschlüssel als vertrauenswürdiger Einführer bestätig) als vertrauenswürdiger Einführer angesehen, das heißt, vollautomatisch erhält dieser Schlüssel den Status „vertrauenswürdig". Das Ganze funktioniert im Endeffekt einfacher, als es sich vielleicht zunächst anhört.

Wenn Sie die Nachfrage, ob die Signierung stattfinden soll, bestätigen, erscheint ein Fenster, in dem Sie das Mantra für den Schlüssel eingeben müssen, mit dem Sie unterschreiben möchten. Wenn Sie mehrere

$\times$ Diesen Schlüssel hat er oder sie auf einem vertrauenswürdigen Weg bekommen – möglichst nicht einfach vom Server! Ein Anruf in der Personalabteilung mit Bitte um den Fingerabdruck ist nicht wirklich sicher, aber besser als gar nichts.

$\diamondsuit$ Hier machen nicht exportierbare Unterschriften Sinn; der Schlüssel der Personalabteilung braucht wirklich nicht von *allen* Mitarbeitern signiert zu sein.

private Schlüssel besitzen, können Sie im oberen Teil durch Klicken auf das Pfeilsymbol eine Auswahlliste mit den vorhandenen Schlüsseln anzeigen lassen und daraus durch Mausklick denjenigen auswählen, den Sie zum Unterschreiben benutzen möchten.

Abbildung 18.17: Schlüssel bereits unterschrieben

Falls der Schlüssel, den Sie unterschreiben möchten, schon einmal mit dem Schlüssel unterschrieben wurde, mit dem Sie es jetzt wieder versuchen, erscheint nach Eingabe und Bestätigung des Mantras eine entsprechende Warnung (Abb. 18.17) und die Prozedur wird abgebrochen.

Wenn Sie ein falsches Mantra eingegeben haben, erscheint ebenfalls eine entsprechende Warnmeldung, nach Bestätigen mit `OK` geht `PGPkeys` wieder zurück zum Mantra-Eingabefenster.

War das Mantra richtig und der Schlüssel nicht bereits schon mit dem Schlüssel unterschrieben, mit dem Sie es jetzt wieder versuchen, dann unterschreibt PGP den Schlüssel nach der Bestätigung ohne weiteren Kommentar.

Anmerkung: Wenn Sie einen RSA-Schlüssel mit einem DSS/ElGamal-Schlüssel signieren, dann kann der RSA-Schlüssel nach dem Export trotzdem in PGP 2.6.x benutzt werden. Die PGP-Versionen vor 5.0 bringen dann eine Warnmeldung, daß ein unbekanntes Dateiformat vorliegt und können mit den DSS/ElGamal-Signaturen nichts anfangen; den RSA-Schlüssel und die evtl. ebenfalls vorhandenen RSA-Signaturen kann PGP 2.6.x trotzdem benutzen. Einstellungen wie „Vertrauenswürdiger Einführer“ gehen dabei aber verloren.

18.5. Anzeigen und Ändern von Schlüssel-Eigenschaften

Mit dem Menübefehl `Keys/Key Properties` (Schlüssel/Eigenschaften) oder durch Mausklick mit der rechten Maustaste und Auswahl von `Key Properties` (Schlüsseleigenschaften) aus dem erscheinenden Kontextmenü können Sie sich die Eigenschaften eines markierten Schlüssels aus der Liste detailliert anzeigen lassen. In diesem Fenster können Sie auch die Vertrauenseinstellungen eines Schlüssels und das Mantra zu einem privaten Schlüssel ändern. Die angezeigten Informationen werden in den folgenden Abschnitten beschrieben.

Mit Klick auf OK beenden Sie die Anzeige des Fensters mit den Schlüsseleigenschaften unter Übernahme eventueller Änderungen, mit der Schaltfläche Abbrechen verwerfen Sie evtl. gemachte Änderungen. Über Hilfe können Sie weitere Informationen zu den angezeigten Punkten bekommen.

18.5.1. Registerkarte Allgemein (General)

Key ID Die eindeutige Schlüsselkennung, anhand der PGP die Schlüssel für die interne Verarbeitung identifiziert. Näheres hierzu finden Sie in Abschnitt 4.2 auf Seite 20.

Created Das Erzeugungsdatum des Schlüssels.

Key Type Die Schlüsselart (DSS/ElGamal- oder RSA-Schlüssel).

Expires Das Verfallsdatum des Schlüssels, falls eines gesetzt ist; sonst Niemals (Never).

Key Size Die Schlüssellänge in Bit. Dieses Feld steht in PGP 5.0i nicht zur Verfügung.

Cipher Die vom Eigentümer des Schlüssels benutzten symmetrischen Algorithmen (CAST, IDEA, TripleDES; in Zukunft evtl. weitere). Dieses Feld steht in PGP 5.0i nicht zur Verfügung.

Validity Die Gültigkeit des Schlüssels in drei Stufen. Links als Text, rechts als Balken dargestellt (bei PGP 6.0i nur als Balken):

Ungültig (invalid), bedingt gültig (marginal) oder voll gültig (complete) (Siehe auch Abschnitt 7.3 auf Seite 63 und 21.5.2.1 auf Seite 259.)

Trust Das Vertrauen, daß Sie dem Besitzer des Schlüssels hinsichtlich seiner PGP-Signaturen unter Schlüsseln Dritter entgegenbringen. Die Darstellung erfolgt in drei Stufen, links als Text, rechts als Schieberegler (bei PGP 6.0i nur als Schieberegler). Über den Schieberegler können Sie mit der Maus die Einstellungen für das Vertrauen in den jeweiligen Schlüsselinhaber ändern.

Als Einstellmöglichkeiten finden Sie hier: Nicht vertrauenswürdig oder unbekannt (untrusted), bedingt vertrauenswürdig (marginal) und voll vertrauenswürdig (complete bzw. ultimate bei privaten

Schlüsseln mit implizitem Vertrauen (s. u.)). Nähere Erläuterungen finde Sie in Abschnitt 7.3 auf Seite 63 im allgemeinen Teil.

Bei neu importierten Schlüsseln ist das Vertrauen immer auf „untrusted“ gesetzt, da Vertrauenseinstellungen normalerweise nicht exportiert werden und daher nicht übernommen werden (und auch gar nicht übernommen werden sollten). Bei ungültigen bzw. nicht als gültig bekannten (d. h. nicht ausreichend beglaubigten) Schlüsseln läßt sich diese Einstellung (natürlich) nicht ändern. Der Versuch, einem ungültigen Schlüssel höhere Vertrauensgrade zuzuweisen, wird mit einer entsprechenden Meldung abgebrochen, die Vertrauenseinstellungen werden automatisch wieder auf „untrusted“ gesetzt.

Implicit Trust Dieses Optionsfeld kann nur angewählt werden, wenn Sie zum öffentlichen auch den privaten Schlüssel besitzen, also bei eigenen Schlüsseln. Wenn die Option aktiviert ist (Häkchen im Feld), dann ist der Schlüssel automatisch voll gültig und voll vertrauenswürdig, da PGP davon ausgeht, daß der Besitzer eines Schlüssels sich wohl selbst vertrauen wird und nicht versucht, sich selbst einen falschen Schlüssel unterzujubeln.

Fingerprint Der Fingerabdruck, anhand dessen ein Schlüssel eindeutig identifiziert werden kann. Hiermit kann z. B. die Authentizität eines Schlüssels, den Ihnen jemand *persönlich bekanntes* per E-Mail geschickt hat, am Telephon durch Vergleich der Fingerabdrücke überprüft werden. Näheres finden Sie in den Abschnitten 7.1.2 auf Seite 60 und 15.15 auf Seite 135.

Enabled Bei öffentlichen Schlüsseln kann hier durch Klick auf das Feld der Schlüssel aktiviert und deaktiviert werden (siehe Abschnitt 18.6.3.4 auf Seite 201). Wenn das Häkchen angezeigt wird, ist der Schlüssel aktiviert. Bei eigenen Schlüsseln kann diese Einstellung nicht geändert werden.

Change Passphrase Diese Schaltfläche steht nur bei der Anzeige der Eigenschaften privater Schlüssel zur Verfügung. Mit ihr kann das Mantra eines Schlüssels geändert werden. Wenn Sie auf diese Schaltfläche klicken, erscheint ein Fenster, in dem Sie das Mantra abändern können (Abb. 18.18). Hierzu müssen Sie im oberen Feld das alte Mantra und darunter das neue Mantra zweimal eingeben, um Tippfehler auszuschließen (zum Mantra allgemein siehe 7.4 auf Seite 65 und 5.1 auf Seite 34). Bei PGP 6.0 müssen Sie zuerst in einem getrennten Fenster das alte Mantra eingeben, wenn dieses richtig eingegeben wurde, erscheint ein zweites Fenster, in dem Sie Ihr neues Mantra zweimal eingeben müssen.

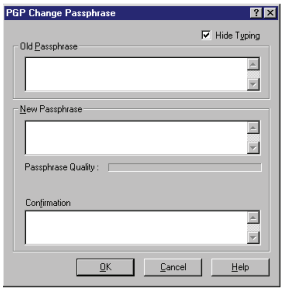


Abbildung 18.18: Mantra ändern (5.5.3i)

18.5.2. Registerkarte Unterschlüssel (Subkeys) (nur PGP 6.0i)

PGP 6.0i bietet die Möglichkeit, einem Schlüssel mehrere Unterschlüssel zuzuordnen, die getrennt vom Masterkey zurückgerufen und ersetzt werden können. In diesem Fenster wird eine Liste dieser Unterschlüssel angezeigt, Sie können hier neue Unterschlüssel erzeugen (mit Klick auf Schaltfläche `New`), bestehende zurückrufen (mit Klick auf Schaltfläche `Revoke`) und schließlich löschen (mit einem Klick auf die Schaltfläche `Remove`). Weitere Anmerkungen zu Unter- oder Teilschlüsseln finden Sie in Abschnitt 4.6 auf Seite 30.

18.5.3. Registerkarte Rückrufer (Revokers) (nur PGP 6.0i)

PGP 6.0i bietet die Möglichkeit, einem Schlüssel einen oder mehrere *Rückrufer* hinzuzufügen, das heißt, einer anderen Person das Recht einzuräumen, den eigenen Schlüssel für ungültig zu erklären und zurückzurufen.

In diesem Dialogfeld sehen Sie eine Liste mit zum Rückruf des Schlüssels berechtigten Schlüsseln. Ändern können Sie an dieser Liste nichts, auch an Ihrem eigenen privaten Schlüssel können Sie einmal eingerichtete Rückrufer nicht wieder löschen (bitte lesen Sie hierzu auch Abschnitt 18.6.3.4 auf Seite 197).

18.6. Übersicht über PGPkeys

18.6.1. Die Schlüsselliste

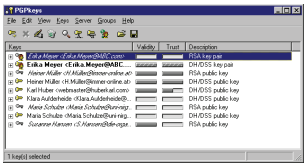


Abbildung 18.19: Die Schlüsselliste in PGPkeys (6.0i)

Im Fenster von PGPkeys werden Ihnen in einer Liste alle Schlüssel angezeigt, die in Ihren Schlüsselbund-Dateien enthalten sind (Abb. 18.19). Ein Schlüssel besteht aus mehreren Teilen: Der eigentliche Schlüssel; ein oder mehrere Benutzernamen (z. B. für unterschiedliche E-Mail-Adressen ein und derselben Person) sowie Unterschriften, das heißt Beglaubigungen von anderen Personen. Diese zusätzlichen Benutzernamen und Beglaubigungen können Sie sich anzeigen lassen.

In der *Listenansicht* von PGPkeys können Sie die zu einem Schlüssel gehörigen Benutzernamen und Unterschriften ein- oder ausblenden. Dazu gibt es mehrere Möglichkeiten. Sie können einen oder mehrere Schlüssel in der Liste per Mausklick markieren und dann die erweiterte Ansicht über das Menü Edit/Expand Selection öffnen bzw. über Edit/Collapse Selection wieder schließen. Wenn Sie auf ein geschlossenes Element doppelklicken, wird es geöffnet, wenn Sie auf ein bereits geöffnetes Element doppelklicken, wird es wieder geschlossen. Als dritte Möglichkeit schließlich können Sie über einen Mausklick in die den Elementen vorangestellten Kästchen mit Plus- oder Minuszeichen die nächste Ebene öffnen bzw. eine Ebene wieder schließen.

In der ersten Spalte unter der Überschrift Keys werden Ihnen die Schlüssel in Ihrem Schlüsselbund angezeigt.

Rechts neben den im letzten Absatz beschriebenen Kästchen zum Öffnen und Schließen der zusätzlichen Ebenen sehen Sie ein Symbol, das Ihnen Aufschluß über die Art des jeweiligen Schlüssels gibt. Ein blauer Schlüssel in Form eines einfachen Schlüssels wie an einer Zimmertür weist Sie darauf hin, daß es sich um einen RSA-Schlüssel handelt. Ein gelber Schlüssel in Form eines Sicherheitsschlüssels steht für einen DSS/ElGamal-Schlüssel.⊗

Wenn das Symbol zwei statt einen Schlüssel darstellt, besitzen Sie auch den privaten Schlüssel, es handelt sich also um Ihren oder einen

⊗ Dies soll nicht bedeuten, RSA sei unsicherer. Es sind nur verschiedene Symbole für verschiedene Schlüsseltypen.

Ihrer persönlichen Schlüssel. Bei PGP 6.0i wird ein persönlicher Schlüssel nicht durch zwei Schlüssel, sondern durch einen Schlüssel mit einem stilisierten Kopf gekennzeichnet.

Ist das Symbol grau, dann ist der Schlüssel deaktiviert, kann also nicht benutzt werden.

Benutzernamen werden durch ein Briefumschlags-Symbol gekennzeichnet (bei PGP 5.0i durch einen stilisierten Kopf), blau bei RSA- und gelb bei DSS/ElGamal-Schlüsseln.

Unterschriften (Beglaubigungen) unter dem Schlüssel werden, unabhängig ob RSA- oder DSS/ElGamal-Schlüssel zur Unterschrift genommen wurden, durch ein Bleistift-Symbol dargestellt (bei PGP 5.0i eine Feder). Wenn es sich um eine spezielle Signatur handelt (Trusted Introducer oder Meta-Introducer, siehe Abschnitt 18.4 auf Seite 181), wird statt des Bleistifts eine Tintenfeder als Symbol angezeigt (bei PGP 5.0i sind diese speziellen Funktionen noch nicht vorhanden, sie werden darum als normale Unterschriften angezeigt).

Rechts neben dem jeweiligen Symbol steht bei Schlüsseln der Standard-Benutzername, bei zusätzlichen Benutzernamen die jeweilige Bezeichnung und bei Unterschriften der Standard-Benutzername des Schlüssels, mit dem unterschrieben wurde. Wenn dieser Benutzername bei einer Unterschrift nicht bekannt ist (wenn sich der Schlüssel des Unterschreibenden also nicht in Ihrem Schlüsselbund befindet), dann steht statt des Benutzernamens dort die Schlüssel-Nummer mit einem Vermerk `unknown signer` oder `unavailable`, also „Unterschreibende unbekannt“ oder „Nicht verfügbar“.

In der zweiten Spalte sehen Sie unter der Überschrift `Validity` Angaben zur Gültigkeit, das heißt zur Frage, wie sicher Sie darüber sein können, daß der Schlüssel auch tatsächlich zu der Person gehört, die sein Benutzername ausgibt. Näheres hierzu finden Sie in Abschnitt 7.3 auf Seite 63. Diese Anzeige ist abhängig von den Einstellungen in den PGP-Grundeinstellungen (siehe Abschnitt 21.5.2.1 auf Seite 259). Sie ist entweder eine einfache Ja-Nein-Anzeige in der folgenden Form (nicht bei PGP 5.0i):

Grauer Runder Punkt Ungültig (nicht oder nur von unbekannten oder nicht vertrauenswürdigen Schlüsseln beglaubigt).

Grüner Runder Punkt Gültig (ausreichend beglaubigt).

Grüne Raute Eigener Schlüssel (daher implizit ausreichend).

Oder sie zeigt die Gültigkeit abgestuft in Form eines Balkens in der folgenden Form an (alle Versionen):

Hellgrau Ungültig (nicht oder nur von unbekannten oder nicht vertrauenswürdigen Schlüsseln beglaubigt).

Halb dunkel-, halb hellgrau Grenzfall (von einem bedingt vertrauenswürdigen Schlüssel beglaubigt).

Dunkelgrau Gültig (ausreichend beglaubigt).

Dunkelgrau mit Streifen Eigener Schlüssel (daher implizit ausreichend beglaubigt).

In der dritten Spalte wird unter der Überschrift `Trust` (Vertrauen) Ihre Einschätzung der Vertrauenswürdigkeit des jeweiligen Schlüsselinhabers hinsichtlich der Schlüsselbeglaubigung mit einem Balken abgestuft dargestellt (zur Bedeutung sei hier noch einmal auf Abschnitt 7.3 auf Seite 63 verwiesen):

Hellgrau Kein Vertrauen in Beglaubigungen mit diesem Schlüssel vorhanden.

Halb dunkel-, halb hellgrau Beglaubigungen mit diesem Schlüssel sind „meistens“ vertrauenswürdig.

Dunkelgrau Beglaubigungen mit diesem Schlüssel sind voll vertrauenswürdig.

Dunkelgrau mit Streifen Für eigene Schlüssel. Volles Vertrauen. PGP geht davon aus, daß Sie wenigstens sich selbst vertrauen können . . .

Im weiteren unterscheiden sich die verschiedenen Versionen etwas. PGP 5.0i zeigt in der vierten Spalte das Erzeugungsdatum der Schlüssel, Benutzernamen oder Unterschriften an und in der fünften Spalte die Schlüssellänge. Hier können die anzuzeigenden Spalten nicht geändert werden.

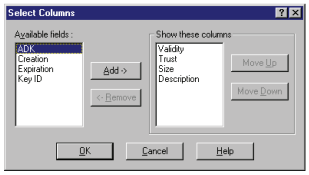


Abbildung 18.20: Auswahl der anzuzeigenden Spalten (5.5.3i)

Bei PGP 5.5.3i wird in der vierten Spalte standardmäßig die Schlüssellänge der jeweiligen Schlüssel angezeigt und in der fünften Spalte unter der Überschrift `Description` (Beschreibung) ein kurzer Beschreibungstext aufgeführt, was in der jeweiligen Zeile enthalten ist. Diese Informationen sind in den Symbolen in der ersten Spalte auch schon enthalten, hier aber etwas leichter verständlich. Bei Version 5.5.3i kann über den Menüpunkt `Keys/Select Columns` (Schlüssel/Spalten auswählen) die Konfiguration der Spalten geändert werden (Abb. 18.20). In der linken Spalte unter `Available Fields` (Zur Verfügung stehende Felder) sehen Sie die zur Verfügung stehenden Spalten, die derzeit nicht angezeigt werden, rechts sehen Sie die bereits oben beschriebenen Spalten. Sie können nun weitere Spalten hinzufügen, indem Sie sie in der linken Spalte mit Mausklick markieren und dann auf die Schaltfläche `Add` (Hinzufügen) klicken. Nicht benötigte Spalten können Sie ausblenden, indem Sie sie in der rechten Spalte markieren und anschließend auf die Schaltfläche `Remove` (Entfernen) klicken. Die Reihenfolge der Spalten können Sie ändern, indem Sie eine anzuzeigende Spalte in der rechten Liste markieren und dann mit Klick auf die Schaltfläche `Move Up` (Nach oben) weiter nach vorne oder mit Klick auf die Schaltfläche `Move Down` (Nach unten) weiter nach hinten bringen. Ihre geänderte Anzeige bestätigen Sie mit `OK`; wenn Sie die Änderungen nicht übernehmen möchten, können Sie mit `Cancel` abbrechen.

Folgende Spalten stehen Ihnen zusätzlich zu den oben beschriebenen für die Anzeige zur Verfügung:

ADK Erzwungene Drittkey-Verschlüsselung; Wenn ein Schlüssel ein ADK-Feld enthält (Additional Decryption Key, bedeutet dies, daß bei Verschlüsselung an diesen Schlüssel automatisch auch an einen Dritten mitverschlüsselt wird, auch bekannt als ARR oder CAK, näheres siehe Abschnitt A.2 auf Seite 267), so wird das durch die Farbe des Punktes angezeigt. Ein grauer Punkt bedeutet, daß kein ADK gesetzt ist, das heißt, daß nur die eigentliche Empfängerin die Nachricht wieder entschlüsseln kann. Diese Anzeige sollte unserer Meinung nach eingeschaltet werden, damit Sie frühzeitig sehen können, daß noch jemand anders die E-Mail lesen kann,

die Sie an diesen Empfänger verschlüsseln. Dafür müssen Sie das Fenster allerdings etwas breiter ziehen.

Expiration Verfallsdatum des Schlüssels, an dem der Schlüssel ungültig wird. Steht auf Never (Niemals), wenn kein Verfallsdatum gesetzt ist.

Key ID Die eindeutige Kennung des Schlüssels bzw. bei einer Signatur des Schlüssels, mit dem unterschrieben wurde. Anhand dieser eindeutigen Schlüsselkennung nimmt PGP die Verknüpfungen zwischen Benutzernamen und Schlüsseln vor. Näheres finden Sie in Abschnitt 4.2 auf Seite 20.

Creation Datum und Uhrzeit der Erzeugung des Schlüssels oder der Unterschrift.

Bei PGP 6.0i wird in der vierten Spalte standardmäßig unter der Überschrift Description ein kurzer Beschreibungstext aufgeführt, was in der jeweiligen Zeile enthalten ist. Diese Informationen sind in den Symbolen in der ersten Spalte auch schon enthalten, hier aber etwas leichter verständlich. In der fünften Spalte steht standardmäßig die Schlüsselnummer (Key-ID, s. o.). Auf Wunsch stehen die selben Spalten zur Anzeige zur Verfügung, wie bereits in den Absätzen bei PGP 5.5.3i beschrieben. Auswählen können Sie die anzuzeigenden Spalten über das Menü View (Ansicht). Ein Häkchen vor dem jeweiligen Spaltennamen bedeutet, daß die Spalte angezeigt wird. Die Anzeigereihenfolge können Sie ändern, indem Sie mit der Maus in den Spaltentitel klicken und mit gedrückter Maustaste die Spalte auf die gewünschte Position ziehen und dort die Maustaste loslassen.

In allen Versionen können Sie die angezeigten Schlüssel nach Benutzernamen, Gültigkeit, Vertrauensniveau, Schlüssellänge, Erstellungsdatum, Verfallsdatum und Schlüsselnummer sortieren, indem Sie mit der Maus einmal in den Spaltentitel klicken. Ein weiterer Mausklick in den jeweiligen Spaltentitel kehrt die Sortierreihenfolge um.

18.6.2. Gruppen (nicht bei PGP 5.0i)

PGP bietet seit der Version 5.5 die Möglichkeit, Empfängerinnen in Gruppen zusammenzufassen, wenn immer wieder an eine gleichbleibende Liste (Vorstand, Projektgruppe, ...) verschlüsselt werden soll. Den

Gruppen werden dann die Schlüssel der zugehörigen Personen zugeordnet. Statt nun jedesmal mehrere Empfänger angeben zu müssen, reicht es aus, die Gruppe anzugeben. Die Daten werden dann automatisch an alle Empfängerinnen verschlüsselt, die Mitglieder der Gruppe sind. Verschlüsselung an ein einzelnes Mitglied der Gruppe, ohne an die anderen Mitglieder zu verschlüsseln, ist natürlich trotzdem möglich, auch wenn ein Schlüssel zu einer Gruppe gehört. Es handelt sich lediglich um eine Kurzschreibweise, um Ihnen die Empfängerwahl zu erleichtern.

18.6.2.1. Anlegen einer neuen Gruppe (nicht bei PGP 5.0i)

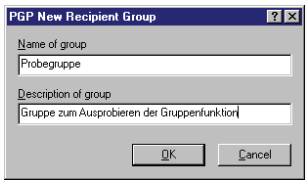


Abbildung 18.21: Anlegen einer neuen Gruppe (5.5.3i)

Mit dem Menübefehl `Group/New Group` wird eine neue Gruppe angelegt. Es erscheint ein Fenster, in dem Sie in der oberen Zeile einen Gruppennamen (in PGP 6.0i Adresse genannt) und im unteren Feld eine Bezeichnung (z. B. eine nähere Erläuterung der Zusammensetzung) eingeben müssen (Abb. 18.21). Mit `OK` bestätigen Sie das Anlegen der Gruppe, mit `Cancel` brechen Sie den Vorgang ohne Anlegen einer neuen Gruppe ab.

18.6.2.2. Gruppen anzeigen (nicht bei PGP 5.0i)

Mit dem Menübefehl `Groups/Show Groups` können Sie die Anzeige der definierten Gruppen in `PGPkeys` an- und ausschalten. Ein Häkchen vor dem Befehl im Menü zeigt an, daß die Gruppenanzeige aktiviert ist. In diesem Fall ist das Anzeigefenster von `PGPkeys` zweigeteilt, im unteren Teil werden die Gruppen mit den jeweiligen Mitgliedern angezeigt (Abb. 18.22). Die Bedienung funktioniert wie bei der Schlüssellisten (siehe Abschnitt 18.6.1 auf Seite 187). In der Liste wird eine Gruppe mit der Gültigkeit des am schwächsten beglaubigten Schlüssels bewertet. Wenn also mindestens ein ungültiger Schlüssel in der Gruppe enthalten ist, wird die gesamte Gruppe als ungültig markiert.

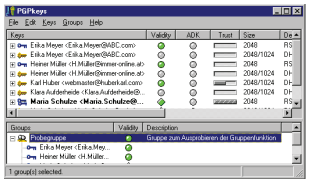


Abbildung 18.22: Schlüsselliste mit Gruppen (5.5.3i)

18.6.2.3. Neue Schlüssel zu einer Gruppe hinzufügen (nicht bei PGP 5.0i)

Neue Schlüssel können Sie zu einer Gruppe hinzufügen, indem Sie den Schlüssel in der Schlüsselliste von PGPkeys (oberer Teil) mit einem Mausklick markieren, die Maustaste gedrückt halten, den Schlüssel mit gedrückter Maustaste auf das Symbol für die Gruppe ziehen und dort wieder loslassen.

Eine andere Möglichkeit ist das Kopieren eines Schlüssels in die Windows-Zwischenablage mit Edit/Copy (Bearbeiten/Kopieren) oder Mausklick mit rechter Maustaste auf den einzufügenden Schlüssel und Auswählen von Copy aus dem Kontextmenü. Dann markieren Sie per Mausklick die Gruppe, in die der Schlüssel aufgenommen werden soll und wählen aus dem Menü den Befehl Edit/Paste (Bearbeiten/Einfügen) oder rufen mit der rechten Maustaste das Kontextmenü auf und wählen den Befehl Paste into Group (In Gruppe einfügen) aus.

18.6.2.4. Schlüssel aus einer Gruppe entfernen/Gruppen löschen (nicht bei PGP 5.0i)

Mitglieder aus einer Gruppe entfernen können Sie, indem Sie den betreffenden Schlüssel in der Gruppenliste (nicht in der Schlüsselliste!) markieren und dann den Befehl Edit/Delete (Bearbeiten/Löschen) auswählen oder mit der rechten Maustaste in der Gruppenliste auf das entsprechende Mitglied klicken und aus dem erscheinenden Kontextmenü den Punkt Remove auswählen. Sie können auf diese Weise auch mehrere Mitglieder auf einmal aus der Gruppe entfernen.

Es erscheint ein Fenster mit einer Sicherheitsabfrage, ob Sie das Mitglied bzw. die Mitglieder aus der Gruppe wirklich entfernen möchten.

Wenn Sie eine Gruppe insgesamt löschen möchten, markieren Sie statt eines oder mehrerer Mitglieder die Gruppe selbst und verfahren wie oben beschrieben.

18.6.2.5. Gruppen importieren (nicht bei PGP 5.0i)

Sie können eine Datei importieren, die die Mitglieder einer Gruppe enthält. Auf diese Weise reicht es in einer geschlossenen Gruppe aus, wenn ein Mitglied die Gruppe erstellt und die Datei an die anderen weitergibt, die Sie dann importieren können. Hierzu wählen Sie den Menübefehl

Groups/Import Groups und wählen in dem erscheinenden Windows-Datei-öffnen-Dialog die entsprechende Datei aus.

Die Datei enthält nicht die eigentlichen Schlüssel der Gruppenmitglieder, sie enthält nur eine Verknüpfung von eindeutiger Schlüsselnummer mit der Gruppe. Die öffentlichen Schlüssel der Gruppenmitglieder müssen Sie sich ggf. getrennt besorgen und sie in Ihren Schlüsselbund aufnehmen. Wenn in einer importierten Gruppe ein Gruppenmitglied existiert, dessen Schlüssel Sie nicht in Ihrem Schlüsselbund haben, zeigt Ihnen PGP in der Mitgliederliste nur eine unbekannte Benutzerin (unknown user) und die eindeutige Schlüsselnummer des fehlenden Schlüssels an. An dieses Mitglied kann dann natürlich nicht verschlüsselt werden.

18.6.2.6. Die Eigenschaften einer Gruppe ändern (nicht bei PGP 5.0i)

Über den Menübefehl Group Properties erhalten Sie ein Fenster, in dem Sie den Gruppennamen und die Bezeichnung der Gruppe ändern können. Das Fenster ist dasselbe wie im Abschnitt 18.6.2.1 auf Seite 192 beschrieben.

18.6.3. Die Menübefehle von PGPkeys

18.6.3.1. Menü Datei (File)

Teile eines geteilten Schlüssels senden (Send Key Shares) (nur PGP 6.0i)

PGP 6.0i bietet die Möglichkeit, einen privaten Schlüssel aufzuteilen und an mehrere Personen zu verteilen. Dann kann der Schlüssel nur verwendet werden, wenn die bei der Aufteilung vorgegebene Anzahl an Teilschlüsseln zusammenkommt. Hierzu müssen die Schlüsselteile zusammengeführt werden. Dies kann entweder auf einem Rechner lokal (dann müssen aber alle betreffenden Personen anwesend sein, um ihr jeweiliges Mantra einzutippen) oder über eine Netzwerkverbindung geschehen. Dies kann z. B. in Firmen sinnvoll sein, wenn nur mehrere Leute gemeinsam berechtigt sind, den Firmen-Schlüssel zu verwenden. Wenn die Teile eines aufgespaltenen Schlüssels über eine Netzwerkverbindung (IP-Netzwerk) zusammengeführt werden sollen, können die Schlüssel-

teile mit dieser Funktion an den Rechner geschickt werden, auf dem die Schlüsselzusammenführung angestoßen wurde.

Achtung: Diese Funktion stellt keine sichere Methode dar, das Gewünschte zu erreichen. Erstens wird der Schlüssel ganz normal erzeugt, kann also vor dem Verteilen kopiert werden; zweitens wird bei der gemeinsamen Benutzung des Schlüssels keine verteilte Berechnung durchgeführt, sondern die Schlüsselteile werden auf einem Rechner zusammengeführt und dort verwendet – wenn dieser Rechner (und sein Benutzer) nicht vertrauenswürdig ist, besteht die Gefahr, daß dort etwas ganz anderes signiert wird als die Schlüsselteilhaber glauben oder gar daß der zusammengesetzte Schlüssel einfach gespeichert wird.

Beenden (Exit)

Beendet PGPkeys ohne Rückfrage.

18.6.3.2. Menü Bearbeiten (Edit)

Kopieren (Copy)
Einfügen (Paste)
Löschen (Delete)

Die unter Windows allgemein üblichen Funktionen Kopieren und Einfügen. Hiermit können Sie (auf dem Umweg über die Zwischenablage) z. B. Schlüssel in Textform in eine E-Mail kopieren.

Kopieren (Copy) ist nur wählbar, wenn Sie einen Schlüssel markiert haben, nicht bei Benutzernamen und Signaturen.

Einfügen (Paste) ist nur wählbar, wenn sich Daten in der Windows-Zwischenablage befinden. Wenn der Befehl angewählt wird, werden die Daten in der Zwischenablage nach Schlüsseln durchsucht. Wenn Schlüsseldaten gefunden werden, startet der Dialog zum Schlüsselimport (siehe Abschnitt 18.2 auf Seite 173). Wenn keine Schlüsseldaten in der Zwischenablage gefunden werden, wird eine entsprechende Meldung angezeigt und der Vorgang abgebrochen.

Löschen (Delete) ist immer wählbar, wenn Sie einen Schlüssel, einen Benutzernamen oder eine Signatur gewählt haben. Sie können aus einem Schlüssel also auch einzelne Signaturen und Benutzernamen löschen. Gehen Sie mit dieser Funktion vorsichtig um, einmal gelöschte Benutzernamen können Sie einem Schlüssel nur dann wieder hinzufü-

gen, wenn es sich um Ihren eigenen Schlüssel handelt. Ansonsten müssen Sie den Schlüssel neu importieren; es ist hierfür nicht nötig, ihn zuvor aus Ihrem Schlüsselbund zu löschen.

Alle auswählen (Select all)
Alle zusammenklappen/Auswahl zusammenklappen (Collapse all/Collapse selection)
Alle erweitern/Auswahl erweitern (Expand all/Expand selection)

Diese Befehle dienen der Markierung und der Wahl der Darstellung der Schlüssel in der Liste. Mit `Select all` markieren Sie alle Schlüssel in der Liste. Mit `Collapse` schalten Sie auf die Kurzdarstellung (nur Haupt-Benutzernamen der Schlüssel), mit `Expand` auf die ausführliche Darstellung (Schlüssel mit allen Benutzernamen und Signaturen) um (siehe hierzu Abschnitt 18.6.1 auf Seite 187).

Wenn ein oder mehrere Objekte markiert sind, bezieht sich der Befehl auf diese Auswahl, der Menüeintrag lautet dann `Collapse` bzw. `Expand Selection`. Ist in der Liste kein Objekt markiert, lautet der Eintrag `Collapse All` bzw. `Expand All`.

Preferences

Dieser Befehl ruft das Fenster mit den PGP Grundeinstellungen auf. Bitte lesen Sie hierzu in Kapitel 21 auf Seite 241 nach.

18.6.3.3. Menü Ansicht (View) (nur PGP 6.0i)

In diesem Menü können Sie auswählen, welche Spalten PGP Ihnen in der Schlüsselliste anzeigen. Zur Auswahl stehen die im Abschnitt 18.6.1 auf Seite 187 genannten Punkte. Ein Häkchen vor dem jeweiligen Punkt zeigt an, daß der betreffende Punkt angezeigt wird. Wenn Sie einen bereits angezeigten Punkt anwählen, wird er abgewählt und umgekehrt.

Im letzten Punkt in diesem Dialog, `Toolbar` (Werkzeugleiste), können Sie nach demselben System auswählen, ob `PGPkeys` unter den Menüs noch eine Werkzeugleiste mit Symbolen für häufig genutzte Befehle anzeigen soll oder nicht.

18.6.3.4. Menü Schlüssel (Keys)

Der Inhalt und die Reihenfolge der Befehle in diesem Menü ist in den einzelnen Versionen etwas unterschiedlich. Wenn einzelne Befehle nur in bestimmten Versionen zur Verfügung stehen, ist dies entsprechend vermerkt.

Unterschreiben (Sign)

Mit diesem Befehl leiten Sie den Dialog zum Unterschreiben eines Schlüssels bzw. eines Benutzernamens ein (siehe Abschnitt 18.4 auf Seite 179). Sie können nur Schlüssel mit den dazugehörigen Benutzernamen unterschreiben, keine anderen Unterschriften (das macht auch keinen Sinn). Daher ist dieser Befehl nicht wählbar, wenn Sie nur eine Unterschrift markiert haben.

Als Standardschlüssel setzen (Set as default key)

Mit diesem Befehl machen Sie den gerade markierten persönlichen Schlüssel zu Ihrem Standardschlüssel, der von PGP für Signaturen verwendet wird, wenn Sie nicht im entsprechenden Dialog etwas anderes angeben.

Für die Verschlüsselung ist diese Einstellung nur dann von Bedeutung, wenn die Option `Immer an Standardschlüssel verschlüsseln (Always encrypt to default Key)` in den Grundeinstellungen gesetzt ist (siehe auch gleichnamiger Abschnitt 21.1.1.1 auf Seite 241).

Für die Entschlüsselung und Signaturprüfung hat diese Einstellung keine Bedeutung, hier wird immer der Schlüssel genommen, an den die Daten verschlüsselt oder mit dem die Daten signiert wurden, wenn der betreffende Schlüssel in Ihrem Schlüsselbund enthalten ist.

Dieser Befehl ist nur wählbar, wenn ein eigener Schlüssel markiert ist, nicht bei öffentlichen Schlüsseln.

Benutzernamen hinzufügen (Add Name) Hinzufügen/Benutzernamen, Photo, Rückrufer (Add/Name, Photo, Revoker) (nur PGP 6.0i)

Mit diesem Befehl können Sie einem Schlüsselpaar (also einem eigenen Schlüssel) in allen Versionen einen neuen Benutzernamen (z. B. eine

zweite E-Mail-Adresse) hinzufügen. Sie können das nur bei Schlüsselpaaren tun, deren Mantra Sie kennen. Fremden öffentlichen Schlüsseln können Sie keine Benutzernamen hinzufügen.

Bei PGP 6.0i können Sie außer Benutzernamen auch noch ein kleines Bild zu Ihrem Schlüssel hinzufügen. Wenn Sie den entsprechenden Befehl aus dem Menü wählen, erscheint ein Fenster, in den Sie das Bild über die Drag & Drop-Funktion von Windows einfügen können. Mit `Select File` können Sie auch über ein Fenster `Datei öffnen` eine Grafikdatei auswählen. Erlaubte Dateiformate sind JPEG- und BMP-Bilder. Die zur Verfügung stehende Größe ist 120×144 Pixel. Der reale Nutzen dieser Option ist äußerst zweifelhaft, dafür vergrößert ihr Einsatz Ihren öffentlichen Schlüssel um rund 16 KByte.

Ebenfalls nur bei PGP 6.0i können Sie einem Schlüsselpaar auch einen Rückrufer hinzufügen. Das bedeutet, daß die betreffende Person Ihren Schlüssel für ungültig erklären und zurückrufen kann. Gedacht ist dies für den Fall, daß z. B. ein Mitarbeiter eine Firma verläßt und die Firma dessen Schlüssel mit dem Firmenschlüssel für ungültig erklären kann oder für den Fall, daß Sie Ihren Schlüssel verloren haben und keine Sicherungskopie haben (was eigentlich nicht passieren sollte – haben Sie eigentlich ein lesbares Backup Ihres privaten Schlüsselpaares?). In derartigen Fällen (und natürlich auch sonst) kann der Schlüssel durch vorher festgelegte Rückrufer zurückgezogen werden. Der Haken an der Sache ist, daß ein einmal angegebener Rückrufer nicht mehr gelöscht werden kann. Wenn Sie das Vertrauen zu der betreffenden Person verlieren sollten, bleibt Ihnen nichts übrig, als sich ein neues Schlüsselpaar ohne oder mit einem anderen Rückrufer zu erzeugen.

Schlüssel, die diese Bilddaten oder einen Rückrufer enthalten, sind nur für Benutzer von PGP 6.0 und neuer (auch GnuPG) nutzbar. Wenn Sie mit Benutzerinnen von PGP vor 6.0 kommunizieren (dies wird im allgemeinen der Fall sein), dann müssen Sie darauf achten, beim Exportieren Ihres Schlüssels für die Weitergabe diese Funktionen abzustellen. (Siehe hierzu den Abschnitt 21.5.2.4 auf Seite 260.)

Bei PGP 5.0i und 5.5.3i heißt der Menübefehl `Add Name`, bei PGP 6.0i heißt er `Add`; wenn Sie `Add` auswählen, erscheint ein Untermenü mit den Auswahlpunkten `Name`, `Photo` und `Revoker`.

Keyserver/Ausgewählte Schlüssel laden, Ausgewählte Schlüssel senden, Suchen (Keyserver/Get Selected Keys, Send Selected Keys, Find New Keys) (nur PGP 5.0i)
Aktualisieren vom Keyserver (Update from Server) (nur PGP 5.5.3i)
Schlüssel zum Server schicken (Send Key to Server) (nur PGP 5.5.3i)
Search (nur PGP 5.5.3i)

Der Befehl `Keyserver/Get Selected Keys` (Ausgewählte Schlüssel vom Server laden) (bei PGP 5.0i) bzw. `Update from Server` (Vom Server aktualisieren) (bei PGP 5.5.3i) holt sich eine aktuelle Kopie des oder der in der Auswahl markierten Schlüssel von dem zuständigen Keyserver (siehe auch Abschnitt 21.4 auf Seite 251).

Der Befehl `Keyserver/Send Selected Keys` (Ausgewählte Schlüssel zum Server schicken) (bei PGP 5.0i) bzw. `Send Key to Server` (Schlüssel zum Server schicken) (bei PGP 5.5.3i) schickt eine Kopie des oder der in der Auswahl markierten Schlüssel mit Benutzernamen und Signaturen an den zuständigen Keyserver (siehe auch Abschnitt 21.4 auf Seite 251).

Der Befehl `Keyserver/Find New Key` (Schlüssel auf Server suchen) (bei PGP 5.0i) bzw. `Search` (Suchen) (bei PGP 5.5.3i) startet die Suchanfrage nach einem neuen, in Ihrem Schlüsselbund nicht vorhandenen Schlüssel auf dem Keyserver.

Bei PGP 5.0i geben Sie im erscheinenden Fenster nur einen Benutzernamen oder eine E-Mail-Adresse des Schlüsselinhabers ein, da Sie nur einen Keyserver einstellen können.

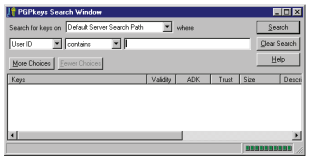


Abbildung 18.23: Suche eines Schlüssels auf dem Server (5.5.3i)

Bei PGP 5.5.3i erscheint ein komplexeres Fenster (Abb. 18.23), das Ihnen vielfältige Auswahlmöglichkeiten für die Suche nach einem Schlüssel gibt. Über das oberste Auswahlfeld können Sie angeben, wo nach dem Schlüssel gesucht werden soll. Zur Auswahl stehen dabei der Standard-Keyserver-Pfad, Ihr lokaler Schlüsselbund, die aktuellen Suchresultate (zur weiteren Bearbeitung einer Anfrage, die zu viele Antworten ergeben hat, was zum Beispiel bei einer Suche nach einem Peter Meyer durchaus der Fall sein könnte) sowie jeder einzelne der in Ihrer Liste eingetragenen Server, bei dem in den Grundeinstellungen das Attribut `Listed` gesetzt wurde (siehe Abschnitt 21.4 auf Seite 251).

Im linken Listenfeld der zweiten Zeile können Sie angeben, wonach Sie suchen möchten. Zur Auswahl stehen: Benutzernamen (User ID), eindeutige Schlüsselnummer (Key ID), Schlüsselart (Key Type), Erzeugungsdatum (Creation Date), Verfallsdatum (Expiration Date), Schlüsselstatus (Key Status) und Schlüssellänge (Key Size).

Im Feld rechts daneben können Sie eine Bedingung angeben. Zur Auswahl stehen bei Benutzernamen z. B. die Bedingungen ist (is) und ist nicht (is not), enthält (contains) und enthält nicht (does not contain), ist unterschrieben von (is signed by) und ist nicht unterschrieben von (is not signed by). Bei den Erzeugungs- oder Verfallsdaten stehen die Bedingungen ist am (is on), ist am oder vor (is on or before), ist am oder nach (is on or after) zur Auswahl, bei den Schlüssellängen die Bedingungen ist (is), ist mindestens (is at least) und ist höchstens (is at most).

Im ganz rechten Feld können Sie entweder einen Text eingeben (bei User ID und Key ID) oder Sie erhalten ein zum Typ des Suchkriteriums passendes Auswahlfeld, das Ihnen z. B. beim Schlüsseltyp die Möglichkeiten Diffie-Hellman⊙ für DSS/ElGamal- und RSA für RSA-Schlüssel anzeigt.

Mit diesen Auswahlfeldern können Sie praktisch jede für Schlüssel relevante Suchbedingung erzeugen wie z. B. „Die Benutzerkennung enthält Maria.Schulze@uni-nirgendwo.de“. Wenn Sie als Suchkriterium die E-Mail-Adresse der Person haben, deren Schlüssel Sie suchen, werden Sie meistens kein weiteres Suchkriterium benötigen. Nur wenn die betreffende Person die E-Mail-Adresse nicht in die Benutzerkennung aufgenommen hat, müssen Sie möglicherweise noch über weitere Kriterien suchen. Aber es ist natürlich auch denkbar, daß Sie nur eine Schlüsselnummer haben (z. B. bei einer Unterschrift mit einem Schlüssel, der nicht in Ihrem Schlüsselbund enthalten ist).

Wenn Sie nach mehr als einem Kriterium suchen möchten, können Sie die Schaltfläche More Choices (Detailliertere Wahl) anklicken, um eine zweite Zeile mit demselben Aufbau wie eben beschrieben zu haben. Wenn das immer noch nicht ausreichen sollte, können Sie auch eine dritte, vierte und fünfte Suchbedingung über diese Schaltfläche hinzufügen. So können Sie Ihre Suche immer weiter verfeinern.

Mit der Schaltfläche Fewer Choices (Gröbere Wahl) nehmen Sie das letzte Kriterium aus der Suchabfrage wieder heraus. Wenn nur noch ein

⊙ Wieder einmal: PGP Inc. nennt ElGamal Diffie-Hellman, was eigentlich ein anderes Verfahren ist.

Suchkriterium übrig ist, kann diese Schaltfläche nicht angewählt werden. Sie können nur das jeweils letzte Suchkriterium löchen.

Mit der Schaltfläche Search (Suche) starten Sie die Suche mit allen angegebenen Kriterien. Angezeigt werden nur Schlüssel, die alle Kriterien erfüllen. Mit der Schaltfläche Clear Search löschen Sie alle eingegebenen Suchkriterien und stellen die Standardeinstellung wieder her.

Neuer Schlüssel (New Key)

Mit diesem Befehl starten Sie das Programm zur Erstellung eines neuen Schlüsselpaares. Bitte lesen Sie hierzu Abschnitt 18.1 auf Seite 166.

Aktivieren (Enable)
Deaktivieren (Disable)

Wenn Sie einen Schlüssel in Ihrem Schlüsselbund behalten möchten, um ihn zu einem späteren Zeitpunkt zu verwenden, ihn aber wegen der Übersichtlichkeit nicht jedesmal in Ihrer Empfängerliste haben möchten, wenn Sie Daten verschlüsseln, dann können Sie ihn in der Liste der Schlüssel markieren und mit Disable deaktivieren. Damit taucht der Schlüssel in der Empfängerliste beim Verschlüsseln von Daten nicht mehr auf. Wenn Sie den Schlüssel wieder benutzen möchten, können Sie ihn markieren und mit dem Befehl Enable wieder aktivieren.

Schlüsselpaare (also eigene Schlüssel) können nicht deaktiviert werden. Ist ein solcher Schlüssel markiert, steht der Menübefehl nicht zur Verfügung. Sie können auch nicht mehrere Schlüssel auf einmal deaktivieren; wenn mehr als ein Schlüssel in der Liste markiert ist, steht der Befehl ebenfalls nicht zur Verfügung.

Deaktivierte Schlüssel werden in der Schlüsselliste durch ein graues Symbol und kursive (schräge) Schrift im Feld Benutzernamen gekennzeichnet.

Zurückrufen (Revoke)

Einen Schlüssel zurückzurufen bedeutet, daß niemand mehr den Schlüssel benutzen soll. Da es keine Möglichkeit gibt, einen Rückruf rückgängig zu machen, sollten Sie diesen Befehl nur ausführen, wenn Sie den betreffenden Schlüssel wirklich nicht mehr benutzen wollen,

weil Sie z. B. den Verdacht haben, daß jemand anderes Zugang zu Ihrem privaten Schlüssel hatte oder dieser Schlüssel aus anderen Gründen unsicher geworden ist.

Sie können nur eigene Schlüssel und Schlüssel, für die Sie als Rückrufer angegeben sind, zurückziehen. Nach einer Sicherheitsrückfrage müssen Sie das zum Schlüssel gehörende Mantra eingeben, um das Zurückziehen des Schlüssels vornehmen zu können.

Wenn Sie den Schlüssel zurückrufen möchten, werden Sie einen triftigen Grund dazu haben. In diesem Fall ist es durchaus angebracht, den Schlüssel mit dem Rückruf an einen Keyserver und an alle Ihre Kommunikationspartner zu schicken, damit sich der Rückruf möglichst schnell verbreitet. Wenn der Rückruf einmal bei Ihrem Gegenüber angekommen ist, kann Ihr Gegenüber mit diesem Schlüssel keine Daten mehr an Sie verschlüsseln.

Bei einem Rückruf ist es normalerweise außerdem sinnvoll, gleich ein neues Schlüsselpaar zu generieren und dieses gleich mit dem Rückruf des alten Schlüssels zu verschicken.

Ein zurückgerufener Schlüssel wird in der Schlüsselliste mit einem durchgestrichenen Symbol gekennzeichnet.

Importieren (Import)

Dieser Menübefehl startet den Dialog zum Import von Schlüssel aus einer Datei (siehe hierzu Abschnitt 18.2.1 auf Seite 174).

Exportieren (Export)

Dieser Menübefehl startet den Dialog zum Export der in der Liste markierten Schlüssel in eine Datei (siehe hierzu Abschnitt 18.3 auf Seite 178).

Schlüssel-Eigenschaften (Key Properties)

Mit diesem Befehl können Sie sich die Eigenschaften eines bestimmten Schlüssels aus der Liste detailliert anzeigen lassen und außerdem die Vertrauenseinstellungen bzw. das Mantra von Schlüsseln ändern (siehe hierzu Abschnitt 18.5 auf Seite 183).

Select Columns (nur PGP 5.5.3i)

In PGP 5.5.3i startet dieser Befehl das Dialogfeld zur Auswahl der von PGPkeys angezeigten Angaben zu den aufgelisteten Schlüsseln (siehe Abschnitt 18.6.1 auf Seite 187).

PGP 5.0i sieht keine Möglichkeit vor, die Anzeige anzupassen, bei PGP 6.0i geschieht dies über das Menü `View` (Siehe Abschnitt 18.6.3.3 auf Seite 196).

Reverify Signatures (nur PGP 6.0i)

Liest alle Signaturen unter allen Schlüsseln im Schlüsselbund neu ein und erstellt die Verknüpfungen (Gültigkeit aufgrund von Vertrauenseinstellungen) neu. Wird normalerweise nicht benötigt.

18.6.3.5. Menü Server (nur PGP 6.0i)

Die Funktionen, die das Verschicken an Keyserver, das Suchen von Schlüsseln auf Keyservern und das Aktualisieren von Schlüsseln mit Schlüsseln von Keyservern angeht, wurden bei PGP 6.0i in ein eigenes Menü gelegt.

Inhaltlich sind die Funktionen mit den in Abschnitt 18.6.3.4 auf Seite 199 erläuterten gleichwertig. Bitte lesen Sie dort nach.

Senden an (Send to) (nur PGP 6.0i)

Bei Auswahl dieses Befehls erscheint ein kleines Untermenü mit den zur Verfügung stehenden Servern. Als oberstes in der Liste erhalten Sie den für die Domain der gesuchten Adresse zuständigen `Domain Server`, darunter die anderen Server in der Liste, sofern Ihr Anzeigeattribut auf `Listed` gesetzt ist (siehe Abschnitt 21.4 auf Seite 251).

Suchen (Search) (nur PGP 6.0i)

Siehe Abschnitt 18.6.3.4 auf Seite 199.

Aktualisieren (Update) (nur PGP 6.0i)

Siehe Abschnitt 18.6.3.4 auf Seite 199.

18.6.3.6. Menü Gruppen (Groups) (nicht bei PGP 5.0i)

Neue Gruppe (New Group) (nicht bei PGP 5.0i)

Mit diesem Befehl wird eine neue Gruppe angelegt. Es erscheint ein Fenster, in dem ein Gruppenname und eine Bezeichnung (z. B. eine nähere Erläuterung der Zusammensetzung) eingegeben werden kann (Abb. 18.21 auf Seite 192). Mit `OK` bestätigen Sie das Anlegen der Gruppe, mit `Cancel` brechen Sie den Vorgang ohne Anlegen einer neuen Gruppe ab (siehe Abschnitt 18.6.2.1 auf Seite 192).

Gruppen anzeigen (Show Groups) (nicht bei PGP 5.0i)

Mit diesem Befehl können Sie die Anzeige der Gruppen in `PGPkeys` an- und ausschalten. Ein Häkchen vor dem Befehl im Menü zeigt an, daß die Gruppenanzeige aktiviert ist. In diesem Fall ist das Anzeigefenster von `PGPkeys` zweigeteilt, im unteren Teil werden die Gruppen mit den jeweiligen Mitgliedern angezeigt (Abb. 18.22 auf Seite 192).

Gruppen importieren (Import Groups) (nicht bei PGP 5.0i)

Mit diesem Befehl können Sie eine Datei importieren, die die Mitglieder einer Gruppe enthält (siehe hierzu Abschnitt 18.6.2.5 auf Seite 193).

Gruppen-Eigenschaften ändern (Group Properties) (nicht bei PGP 5.0i)

Mit diesem Befehl gelangen Sie an ein Fenster, in dem Sie die Eigenschaften einer Gruppe ändern können (siehe hierzu Abschnitt 18.6.2.6 auf Seite 194).

18.6.3.7. Menü Hilfe (Help)

Hilfethemen (Help Topics)

Mit diesem Menübefehl gelangen Sie an die Windows-Online-Hilfe zu PGP. Wie unter Windows üblich, ist die Online-Hilfe in drei Registerkarten eingeteilt. Zur Auswahl stehen:

Inhalt Nach Thematik geordnetes Inhaltsverzeichnis (bei PGP 5.0i nicht verfügbar)

Index Eine alphabetisch geordnete Suchliste mit Themen, die häufiger benötigt werden. Sie können ein Thema direkt eingeben oder aus der Liste auswählen.

Suchen Hier können Sie mit Hilfe einer Wortliste der in der Online-Hilfe enthaltenen Wörter sich die Themen auswählen, in denen Ihr gewünschtes Stichwort vorkommt.

Für die Bedienung der Windows-Hilfe sehen Sie bitte in Ihrer Windows-Dokumentation nach.

PGP Aktualisieren (Upgrade) (nur bei PGP 5.5.3i)

PGP 5.5.3i verfügt über eine eingebaute Aktualisierungsoption über das Internet. Wenn Ihr Internet-Anschluß aktiv und Ihr Windows entsprechend konfiguriert ist, startet diese Funktion Ihren Browser und baut eine Verbindung zu einem Server auf, von wo Sie eine neuere Version von PGP herunterladen können (sofern vorhanden). Bitte lesen Sie bei Problemen die Dokumentation Ihres Browsers.

Über PGP (About PGP)

Bei diesem Menübefehl erscheint ein Fenster, in dem Sie einige kurze Informationen zum Programm und zum Urheber des Programms erhalten. Mit Hilfe der Schaltfläche `Credits` in diesem Fenster können Sie sich eine Liste mit den Namen einiger maßgeblich an der Entwicklung beteiligter Personen anzeigen lassen.

Über eine weitere Schaltfläche können Sie Ihren Browser starten und auf die Internet-Homepage `http://www.PGP.com` (bei PGP 5.5.3i `http://www.PGPi.com`) gelangen.

Mit `OK` bestätigen Sie und gelangen zurück zu `PGPkeys`.

19. PGP benutzen – Aufrufmöglichkeiten

Die Funktionen von PGP werden Ihnen auf mehrere unterschiedliche Arten zur Verfügung gestellt. Wenn Sie die PGP-Funktionen (verschlüsseln, entschlüsseln, signieren, Signatur prüfen) auf Daten in der Windows-Zwischenablage anwenden möchten, so stehen Ihnen hierfür das Menü von `PGPtray` und die entsprechende Funktion von `PGPtools` zur Verfügung. Wenn Sie die PGP-Funktionen (zusätzlich zu den oben genannten noch das Überschreiben und Löschen von Dateien oder ganzen Laufwerken) auf Dateien oder ganze Laufwerke anwenden möchten, so stehen Ihnen hierfür die Windows-Explorer-Erweiterungen und `PGPtools` zur Verfügung.

Für einige E-Mailprogramme gibt es PGP-Erweiterungen, mit denen Sie die Funktionen direkt in Ihrem E-Mailprogramm nutzen können. Wenn Sie keines dieser direkt unterstützten Programme verwenden, so können Sie PGP-Funktionen auf Ihre E-Mails immer noch über den Umweg über die Windows-Zwischenablage anwenden.

Die Verfahrensweise ist für die jeweilige Funktion von PGP immer gleich, gleichgültig, wie Sie sie aufgerufen haben. Daher werden in diesem Kapitel zuerst die verschiedenen Möglichkeiten zum Aufrufen der Funktionen beschrieben und danach die Vorgehensweise für die verschiedenen Funktionen allgemeingültig erläutert.

19.1. Die Explorer-Erweiterungen von PGP

Um auf einem Datenträger abgespeicherte Dateien mit den Funktionen von PGP zu bearbeiten, stehen Ihnen nach der Installation von PGP Erweiterungen für den Windows-Explorer zur Verfügung.

Im Windows-Explorer wird sowohl im Menü `Datei` als auch im Kontextmenü (das beim Drücken der rechten Maustaste erscheint) ein zusätzlicher Befehl `PGP` eingefügt, der seinerseits ein Untermenü mit den PGP-Funktionen öffnet. Der Inhalt dieses Menüs ist abhängig von den Dateitypen der gerade ausgewählten Dateien.

Wenn die markierten Dateien eine der Namensendungen `.pkr` oder `.skr` haben, enthält das PGP-Untermenü den Punkt `Add Keys to Keyring` (Schlüssel zum Schlüsselbund hinzufügen), da Windows davon ausgeht, daß es sich um PGP-Schlüsselbunddateien handelt.

Wenn die Namen markierter Dateien auf `.pgp` enden, dann enthält das PGP-Untermenü den Punkt `Decrypt/Verify` (Entschlüsseln/Überprüfen), da Windows annimmt, daß es sich um verschlüsselte und/oder mit PGP signierte Dateien handelt. Beim Öffnen findet PGP aber auch in solchen Dateien enthaltene Schlüssel und startet den Schlüsselimport, auch wenn dieser Punkt im Menü nicht zur Verfügung stand (siehe Abschnitt 18.2.1 auf Seite 174).

Wenn die markierten Dateien als Suffix `.sig` tragen, dann enthält das PGP-Untermenü den Punkt `Verify signature` (Signatur überprüfen), da PGP diesen Namensbestandteil für abgetrennte Signaturen verwendet.

Wenn die markierten Dateien keine der oben genannten, von PGP bei Windows registrierten Dateiendungen haben, dann enthält das PGP-Untermenü die Befehle `Encrypt` (Verschlüsseln), `Sign` (Signieren) und `Encrypt and Sign` (Verschlüsseln und Signieren).

Für alle Dateien, unabhängig von ihrem Dateityp, enthält das PGP-Untermenü den Befehl `Wipe` (nicht in PGP 5.0i), mit dem das Überschreiben und Löschen der markierten Dateien eingeleitet wird (Siehe hierzu Abschnitt 5.4 auf Seite 37 und 21.1.3 auf Seite 244).

Um eine Datei zu verschlüsseln, genügt es im allgemeinen, sie mit der rechten Maustaste anzuklicken und den Befehl `Encrypt` auszuwählen, um den entsprechenden Vorgang zu starten, der in Abschnitt 20.1 auf Seite 231 näher beschrieben ist. Problematisch wird es nur dann, wenn Sie eine Datei verschlüsseln oder signieren möchten, die bereits einen von PGP registrierten Namen hat, da dann die entsprechenden Funktionen nicht im PGP-Untermenü zur Verfügung stehen. Oder andersherum, wenn Sie eine Datei entschlüsseln möchten, die keine von PGP registrierte Namenserweiterung hat.

In diesen Fällen müssen Sie den Namen der Datei ändern, wenn Sie die Datei über die PGP-Explorer-Erweiterungen bearbeiten möchten. Im allgemeinen wird es einfacher sein, die Bearbeitung der Datei über `PGPtools` einzuleiten (siehe Abschnitt 19.2 auf der nächsten Seite). In PGP 5.0i ist das nicht möglich, da `PGPtools` erst ab PGP 5.5.3i verfügbar ist.

Um den Namen einer Datei ändern zu können, müssen Sie zunächst Windows so einstellen, daß Ihnen der vollständige Name angezeigt wird.

Wählen Sie hierzu im Windows-Explorer unter dem Menü `Ansicht` den Eintrag `Optionen`. In dem dann erscheinenden Fenster müssen Sie auf der Registerkarte `Ansicht` die Option `Keine MS-DOS-Erweiterungen für registrierte Dateien` durch Mausklick auf das Häkchen abschalten. Danach können Sie die Datei inclusive der Erweiterung, anhand derer Windows den Dateityp feststellt, mit den normalen Explorer-Funktionen umbenennen und Ihr damit einen passenden Typ zuweisen.

19.2. `PGPtools` – Schnellstartleiste für PGP-Funktionen (nicht für PGP 5.0i)

`PGPtools` ist ein Programm, das die PGP-Funktionen über ein Fenster mit Schnellstart-Schaltflächen zur Verfügung stellt. `PGPtools` ist nicht für PGP 5.0i verfügbar. Von links nach rechts stehen die folgenden Funktionen zur Verfügung:

Abbildung 19.1: `PGPtools` (6.0i)

19.2.1. Start von `PGPkeys` (nicht für PGP 5.0i)

Durch Mausklick auf das äußerste linke Symbol wird das in Kapitel 18.6 auf Seite 187 beschriebene Programm `PGPkeys` zur Schlüsselverwaltung gestartet. Dies entspricht dem Aufruf von `PGPkeys` über den Eintrag im Windows-Startmenü oder über das Menü von `PGPtray`.

19.2.2. Verschlüsseln (Encrypt) (nicht für PGP 5.0i)

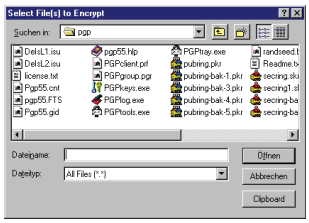


Abbildung 19.2: Auswahl der zu verschlüsselnden Datei (5.5.3i)

Wenn Sie auf die zweite Schaltfläche von links mit der Maus klicken, erscheint ein Fenster, in dem Sie eine oder mehrere Dateien auswählen können, die Sie verschlüsseln möchten (Abb. 19.2). Das Fenster entspricht im Großen und Ganzen einem normalen Windows-Dateiauswahl-Fenster. Zu beachten ist vor allem die Schaltfläche Clipboard, mit der Sie die Verschlüsselung statt auf eine Datei auch auf den Inhalt der Windows-Zwischenablage anwenden können.

19.2.3. Signieren (Sign) (nicht für PGP 5.0i)

Wenn Sie auf die dritte Schaltfläche von links mit der Maus klicken, erscheint ein Fenster, in dem Sie die Datei bzw. Dateien auswählen können, die Sie signieren möchten. Das Fenster ist analog zum Auswahlfenster bei Verschlüsselung (siehe Abschnitt 19.2.2 auf der vorherigen Seite).

19.2.4. Verschlüsseln und Signieren (Encrypt + Sign) (nicht für PGP 5.0i)

Wenn Sie auf die vierte Schaltfläche von links mit der Maus klicken, erscheint ein Fenster, in dem Sie die Datei(en) auswählen können, die Sie verschlüsseln und gleichzeitig signieren möchten. Das Fenster entspricht dem Auswahlfenster bei Verschlüsselung (siehe Abschnitt 19.2.2 auf der vorherigen Seite).

19.2.5. Entschlüsseln/Überprüfen (Decrypt/Verify) (nicht für PGP 5.0i)

Wenn Sie auf die fünfte Schaltfläche von links mit der Maus klicken, erscheint ein Fenster, in dem Sie die Datei bzw. Dateien auswählen können, die Sie entschlüsseln oder deren Signaturen Sie prüfen möchten. Das Fenster ist im wesentlichen dasselbe wie das Auswahlfenster bei Verschlüsselung (siehe Abschnitt 19.2.2 auf der vorherigen Seite).

19.2.6. Überschreiben und Löschen (Wipe) (nicht für PGP 5.0i)

Wenn Sie auf die ganz rechte Schaltfläche (bei PGP 5.5.3i) bzw. die zweite von rechts (bei PGP 6.0i) mit der Maus klicken, erscheint ein Fenster, in dem Sie eine Datei bzw. auch mehrere Dateien auswählen können, die Sie überschreiben und anschließend löschen möchten. Das Fenster entspricht dem Auswahlfenster bei Verschlüsselung (siehe Abschnitt 19.2.2 auf der vorherigen Seite). Zum Thema „Überschreiben und Löschen“ können Sie im Abschnitt 5.4 auf Seite 37 mehr erfahren. Außerdem ist in diesem Zusammenhang Abschnitt 21.1.3 auf Seite 244 interessant.

Achtung: Wenn Sie eine Verknüpfung zum Überschreiben auswählen, so löscht PGP *die Datei, auf die die Verknüpfung zeigt.* Das ist ein berechtigtes Vorgehen (und technisch bedingt auch nicht anders möglich, PGP

erfährt nur den Namen der Datei, auf die die Verknüpfung zeigt), aber Sie sollten daran denken.

19.2.7. Überschreiben und Löschen freier Bereiche (Free Space Wipe) (nur bei PGP 6.0i)

Wenn Sie bei PGP 6.0i auf die ganz rechte Schaltfläche mit der Maus klicken, starten Sie das Programm zum Überschreiben mit Zufallsdaten der unbelegten, also keiner Datei zugeordneten Bereiche auf einem Datenträger. Zum Thema „Überschreiben und Löschen“ können Sie im Abschnitt 5.4 auf Seite 37 mehr erfahren, des weiteren in 21.1.3 auf Seite 244.

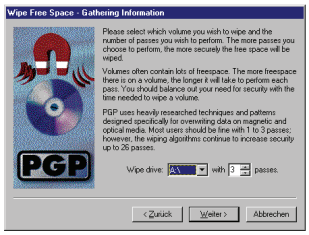


Abbildung 19.3: Freie Bereiche überschreiben (6.0i)

Das erste Fenster, das daraufhin erscheint, enthält eine Warnmeldung und Erklärung des Vorgangs. Mit `Weiter` bestätigen Sie und gelangen zum nächsten Fenster, mit der Schaltfläche `Abbrechen` können Sie den Vorgang ohne Überschreiben beenden.

Wenn Sie nicht abgebrochen haben, erscheint ein Fenster, in dem Sie aus einem Listenfeld das Laufwerk auswählen können, auf dem Sie die ungenutzten Bereiche löschen möchten und in dem Sie angeben können, wie oft PGP die freien Bereiche überschreiben soll (Abb. 19.3). Nach der Bestätigung mit `Weiter` wird Ihnen ein Fenster angezeigt, in dem eine Statistik über das ausgewählte Laufwerk und die Fortschrittsanzeige für das Überschreiben angezeigt werden (Abb. 19.4). Hier ist die letzte Möglichkeit, abzubrechen, bevor der eigentliche Überschreibungsvorgang mit Mausklick auf die Schaltfläche `Begin Wipe` gestartet wird.

Vor dem Überschreiben prüft das Programm den Datenträger auf Fehler im Dateisystem. Werden dabei Fehler gefunden, so wird das Überschreiben nicht durchgeführt, eine entsprechende Fehlermeldung wird angezeigt. In diesem Fall müssen Sie das Laufwerk erst mit einem Reparaturprogramm für Dateisysteme (wie z. B. das mit Windows mitgelieferte `Scandisk`) wieder reparieren, bevor Sie die Funktion ausführen können – die Funktion würde sonst mit an Sicherheit grenzender Wahrscheinlichkeit Daten überschreiben, die Sie noch brauchen.

Abbildung 19.4: Fortschrittanzeige Bereiche löschen (6.0i)

19.3. `PGPtray` – PGP für die Windows-Zwischenablage

`PGPtray` stellt Ihnen die Funktionen von PGP in Form eines Menüs zur Verfügung, das bei einem Mausklick auf das `PGPtray`-Symbol in der rechten Ecke Ihrer Windows-Startleiste angezeigt wird. Die von `PGPtray` zur Verfügung gestellten Funktionen teilen sich in zwei Bereiche auf, nämlich in Funktionen zum Start anderer Programmteile von PGP und in Funktionen, mit denen die in der Windows-Zwischenablage befindlichen Daten mit PGP bearbeitet werden können (verschlüsseln, entschlüssen usw.). Die Reihenfolge der Funktionen im Menü hat sich dabei im Laufe der Zeit verändert. Bei PGP 5.0i und PGP 5.5.3i sind die Funktionen in der untenstehenden Reihenfolge von oben nach unten im Menü aufgeführt, bei PGP 6.0i von unten nach oben.

19.3.1. Daten in der Windows-Zwischenablage verschlüsseln (Encrypt Clipboard)

Dieser Menübefehl startet den in Abschnitt 20.1 auf Seite 231 beschriebenen Verschlüsselungs-Dialog.

19.3.2. Daten in der Windows-Zwischenablage signieren (Sign Clipboard)

Dieser Menübefehl startet den in Abschnitt 20.2 auf Seite 234 beschriebenen Signatur-Dialog.

19.3.3. Daten in der Windows-Zwischenablage verschlüsseln und signieren (Encrypt and Sign Clipboard)

Dieser Menübefehl startet zuerst den in Abschnitt 20.1 auf Seite 231 beschriebenen Verschlüsselungs-Dialog und anschließend den in Abschnitt 20.2 auf Seite 234 beschriebenen Signatur-Dialog – obwohl die tatsächliche Reihenfolge der Aktionen umgekehrt verläuft: Die Daten werden zuerst signiert und dann verschlüsselt, so daß die Signatur nur vom vorgesehenen Empfänger gefunden und geprüft werden kann.

19.3.4. Daten in der Windows-Zwischenablage entschlüsseln oder Signatur überprüfen (Decrypt/Verify Clipboard)

Wenn dieser Menübefehl gewählt wird, überprüft PGP die Windows-Zwischenablage, ob sich darin mit PGP verschlüsselte Daten, PGP-Signaturen oder PGP-Schlüssel befinden. Abhängig von den vorgefundenen Daten werden die entsprechenden Dialoge gestartet. Bitte lesen Sie hierzu in den entsprechenden Abschnitten 20.4 auf Seite 237, 20.5 auf Seite 238 oder 18.2 auf Seite 173 nach.

Wenn die Windows-Zwischenablage keine für PGP relevanten Informationen enthält, also weder verschlüsselte noch signierte Daten und auch keine PGP-Schlüssel, dann wird eine entsprechende Meldung angezeigt und der Vorgang nach Bestätigen der Meldung mit `OK` abgebrochen.

Bei PGP 5.0i kann mit diesem Menübefehl kein Schlüssel aus der Zwischenablage importiert werden; PGP 5.0i zeigt die Meldung an, daß keine relevante PGP-Information gefunden wurde, wenn versucht wird, einen PGP-Schlüssel in der Zwischenablage mit `Decrypt/Verify Clipboard` zu bearbeiten. Bei den neueren PGP-Versionen ab 5.5.3i funktioniert dies. Bei PGP 5.0i müssen Sie Schlüssel mit dem dafür vorgesehenen Menübefehl `Add Key from Clipboard` importieren, der im folgenden Abschnitt beschrieben wird.

19.3.5. Schlüssel aus der Windows-Zwischenablage hinzufügen (Add Key from Clipboard)

Dieser Menübefehl startet den in Abschnitt 18.2.1 auf Seite 174 beschriebenen Dialog zum Importieren von Schlüsseln in Ihren Schlüsselbund. Bei PGP 5.0i können Sie Schlüssel in der Windows-Zwischenablage nur

mit diesem Befehl dem Schlüsselbund hinzufügen, bei den neueren Versionen ab PGP 5.5.3i geht dies auch mit dem im vorangegangenen Abschnitt beschriebenen Befehl.

19.3.6. Text in der Windows-Zwischenablage bearbeiten (Edit Clipboard Text)

Dieser Menübefehl startet einen PGP-eigenen Zwischenablagen-Editor, in dem Ihnen der Inhalt der Windows-Zwischenablage angezeigt wird. Sie können in diesem Fenster den Inhalt der Zwischenablage auch bearbeiten. Mit der Schaltfläche `Save` (bei PGP 5.0i) bzw. `Copy to Clipboard` (PGP 5.5.3i und PGP 6.0i) kopieren Sie den evtl. geänderten Text zurück in die Zwischenablage, Sie übernehmen also die Änderungen. Mit der Schaltfläche `OK` beenden Sie den Editor, *ohne* die Änderungen in die Zwischenablage zu schreiben.

19.3.7. Zugehöriges Anzeigeprogramm starten (Launch Associated Viewer) (nur bei PGP 5.0i)

Dieser Menübefehl startet statt des PGP-eigenen Zwischenablagen-Editors den Windows-Editor (`Notepad.exe`). PGP ist nicht in der Lage, festzustellen, um welche Art Daten es sich beim Inhalt der Zwischenablage handelt. Daher startet es nicht, wie der Name erwarten läßt, das zum Datentyp passende Anzeigeprogramm – es startet einfach immer `notepad`.

Da diese Funktion ziemlich überflüssig ist, wurde Sie in späteren PGP-Versionen weggelassen bzw. durch andere Funktionen ersetzt.

19.3.8. Windows-Zwischenablage leeren (Empty Clipboard) (nicht bei PGP 5.0i)

Dieser Menübefehl löscht den Inhalt der Zwischenablage. Vor dem Löschen erfolgt noch eine Sicherheitsabfrage, die Sie zum Löschen bestätigen müssen.

19.3.9. Aktuelles Fenster benutzen (Use Current Window) (nur bei PGP 6.0i)

Wenn diese Option aktiviert ist (erkennbar am Häkchen vor dem Menüpunkt), werden die Funktionen von `PGPtray` bezüglich Verschlüsselung, Signatur, Entschlüsselung und Signaturprüfung nicht auf den Inhalt der Windows-Zwischenablage angewandt, sondern auf den Inhalt des gerade aktiven Fensters. Dies erspart das vorherige Markieren und das Kopieren des zu behandelnden Textes in die Zwischenablage.

Beim Verschlüsseln und Signieren über `PGPtray` wird, wenn diese Option aktiv ist, nach erfolgter Verschlüsselung bzw. Signatur gleich der verschlüsselte Text bzw. der Klartext und die Signatur anstelle des ursprünglichen Fensterinhalts eingesetzt, der Originaltext wird also überschrieben. Wenn Sie die Daten verschlüsseln, können Sie den Ursprungstext nicht mehr lesen, wenn Sie nicht auch an Ihren eigenen Schlüssel verschlüsselt haben!

Beim Entschlüsseln und Signatur prüfen wird der (verschlüsselte oder signierte) Text im Fenster markiert, die entschlüsselten Daten werden Ihnen aber nicht automatisch eingesetzt, sondern nur im PGP-Zwischenablagen-Editor angezeigt. Das Ergebnis einer Signaturprüfung wird Ihnen in einem separaten Fenster angezeigt. Hierdurch werden die entschlüsselten Daten, die für andere Personen lesbar sind, nur dann auf Ihrem Rechner abgespeichert, wenn Sie dies ausdrücklich veranlassen. Sollten Sie den verschlüsselten Text durch den lesbaren, entschlüsselten Text ersetzen wollen, so müssen Sie im PGP-Zwischenablagen-Editor, der den Klartext enthält, auf die Schaltfläche `Copy to Clipboard` klicken. Daraufhin beendet sich der PGP-Zwischenablagen-Editor und Sie können nun in Ihrem Anwendungsfenster die bereits markierten verschlüsselten Daten durch Wählen von `Einfügen` bzw. `Paste` durch die Klartextdaten ersetzen.

19.3.10. `PGPtools` starten (Launch `PGPtools`) (nicht bei PGP 5.0i)

Mit diesem Menübefehl wird das in Abschnitt 19.2 auf Seite 208 beschriebene Programm `PGPtools` gestartet.

19.3.11. `PGPkeys` starten (Launch `PGPkeys`)

Mit diesem Menübefehl wird das in Kapitel 18.6 auf Seite 187 beschriebene Programm `PGPkeys` gestartet.

19.3.12. `PGPdisk` starten (Launch `PGPdisk`) (nur bei PGP 6.0i)

Dieser Menübefehl ist immer deaktiviert, da `PGPdisk` in der Freeware-Variante von PGP 6.0 nicht enhalten ist. `PGPdisk` ist ein Programm zum Verschlüsseln der Daten auf Ihrer Festplatte.

19.3.13. PGP-Grundeinstellungen (PGP Preferences)

Mit diesem Menübefehl wird das in Kapitel 21 auf Seite 241 beschriebene Fenster zum Ändern der PGP-Grundeinstellungen geöffnet.

19.3.14. Hilfe (Help)

Mit diesem Menübefehl starten Sie die PGP-Online-Hilfefunktion, die in Kapitel 18.6.3.7 auf Seite 205 bereits kurz beschrieben wurde. Zum Gebrauch der Hilfe lesen Sie bitte Ihre Windows-Dokumentation.

19.3.15. `PGPtray` beenden (Quit `PGPtray` bei PGP 5.0i bzw. `Exit PGPtray` bei PGP 5.5.3i und PGP 6.0i)

Mit diesem Befehl beenden Sie das Programm `PGPtray`. Das entsprechende Symbol wird aus Ihrer Windows-Taskleiste entfernt. Falls Sie `PGPtray` erneut starten möchten, können Sie das über den entsprechenden Eintrag im Windows-Startmenü tun.

19.4. Die Programmerweiterungen für E-Mailprogramme (Plugins)

Verschlüsselung macht gerade beim Versenden von Daten über die recht ungeschützten Kommunikationswege in öffentlichen Datennetzen Sinn. Theoretisch kann jeder, der Zugang zu einem Knotenpunkt im Netz hat, alle E-Mails mitlesen, die darüber verschickt werden. Um sich vor unerwünschter Neugier von Dritten zu schützen, gibt es letztendlich nur die

Möglichkeit, die Daten für die Augen Unbeteiligter durch Verschlüsselung unlesbar zu machen.

Es ist bekannt, daß z. B. die Geheimdienste in großem Umfang den Kommunikationsverkehr über öffentliche Netze abhören (Stichworte sind z. B. Echelon, aber auch die Überwachung sämtlicher Auslandstelephonate durch den Bundesnachrichtendienst BND). Aber nicht nur die Geheimdienste, sondern auch jede Menge mehr oder weniger vertrauenswürdige Geschäftemacher und Systembetreuer mit großer Neugier bedrohen Ihre Privatsphäre. Oder ist Ihnen wohl bei dem Gedanken, daß jemand Ihre Liebesbriefe mitliest? Alles schon vorgekommen ...

Gerade für die E-Mail-Kommunikation, das Haupteinsatzgebiet der Verschlüsselung für Privatpersonen und viele Firmen, gibt es die Möglichkeit, die Verwendung von PGP weitgehend zu automatisieren und damit für den täglichen Gebrauch deutlich zu vereinfachen.

Wenn ein PGP-Plugin für Ihr E-Mailprogramm verfügbar ist, so können Sie die Funktionen Verschlüsselung, Signierung, Entschlüsselung und Signaturprüfung weitgehend automatisieren. Wie sich PGP im Zusammenspiel mit Ihrem E-Mailprogramm verhalten soll, können Sie durch die Grundeinstellungen bestimmen. Bitte lesen Sie hierzu Abschnitt 21.1.1 auf Seite 241 und Abschnitt 21.3 auf Seite 248.

19.4.1. PGP-Plugin für Qualcomm Eudora (Light)

Das PGP-Plugin für Qualcomms E-Mailprogramm Eudora (Light) stellt innerhalb von Eudora die PGP-Funktionen zur Verfügung. Für PGP 5.0i muß es gesondert installiert werden, da es nicht im PGP-Programmpaket enthalten ist. Ab PGP 5.5.3i kann das Plugin für Eudora über das PGP-Installationsprogramm zusammen mit PGP installiert werden.

Sie finden das Eudora-Plugin für PGP 5.0i als komprimiertes Zip-Archiv unter dem Dateinamen `PGP50Eud305W95.zip` auf der beiliegenden CD. Nach dem Entpacken mit einem geeigneten Programm, das Zip-Archive entpacken kann (z. B. der Freeware `Alladin Expander`), müssen Sie die Datei `pgpplugin.dll` in das Verzeichnis `plugins` kopieren, das sich seinerseits im Eudora-Programmverzeichnis auf Ihrer Festplatte befindet.

19.4.1.1. Hauptprogrammfenster

Im Menü des Programmfensters von Eudora (Light) wird ein Punkt PGP eingefügt, der die folgenden Befehle enthält (Abb. 19.5):

`PGPkeys` **starten (Launch** `PGPkeys`**)**
Startet das Programm zur Schlüsselverwaltung, `PGPkeys`. Näheres hierzu lesen Sie bitte in Kapitel 18.6 auf Seite 187 nach.

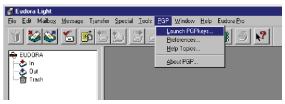


Abbildung 19.5: PGP in Eudora

PGP Grundeinstellungen (Preferences)
Startet das Fenster zum Ändern der PGP Grundeinstellungen. Hierzu lesen Sie bitte in Kapitel 21 auf Seite 241 nach. Insbesondere sei hier auf den Abschnitt 21.3 auf Seite 248 verwiesen, in dem das Verhalten des PGP-Plugins im Zusammenspiel mit dem E-Mailprogramm gesteuert wird.

Hilfe aufrufen (Help Topics) Startet die PGP Online-Hilfe. Bitte lesen Sie hierzu im Abschnitt 18.6.3.7 auf Seite 205 und in Ihrer Windows-Dokumentation nach.

Über PGP (About PGP)
Zeigt ein Fenster mit Informationen über PGP an.

19.4.1.2. Fenster für eingehende E-Mails

Im Fenster einer eingehenden Mail von Eudora Light stehen Ihnen drei zusätzliche Schaltflächen zur Verfügung (von links nach rechts):

Decrypt PGP Encrypted Email Message Wenn auf diese Schaltfläche geklickt wird, wird der Dialog zur Entschlüsselung gestartet. Bitte lesen Sie hierzu im Abschnitt 20.4 auf Seite 237 nach. Dieser Vorgang kann beim Öffnen einer Nachricht automatisch eingeleitet werden. Bitte beachten Sie hierzu Abschnitt 21.3 auf Seite 248.

Wenn Sie in den Grundeinstellungen angegeben haben, daß Ihr Mantra zum Entschlüsseln für eine gewisse Zeit im Speicher gehalten werden soll und diese Zeit seit der letzten Entschlüsselung noch nicht abgelaufen ist, dann müssen Sie das Mantra nicht wieder eintippen. Beachten Sie hierzu bitte die Erläuterungen im Abschnitt 21.1.1.2 auf Seite 243.

Extract PGP Key(s) From Email Message Wenn im E-Mail-Text ein PGP-Schlüssel enthalten ist, wird beim Klicken auf diese Schaltfläche der Schlüsselimport gestartet. Weiteres zum Schlüsselimport lesen Sie bitte im Abschnitt 18.2.3 auf Seite 178 nach. Dieser Vorgang kann beim Öffnen einer Nachricht automatisch eingeleitet werden. Bitte beachten Sie hierzu Abschnitt 21.3 auf Seite 248.

Launch `PGPkeys` Application Startet das Programm zur Schlüsselverwaltung, `PGPkeys`. Hierzu lesen Sie bitte Kapitel 18.6 auf Seite 187.

19.4.1.3. Fenster für ausgehende E-Mails

Im Fenster einer ausgehenden Mail von Eudora Light (Abb. 19.6) stehen Ihnen vier zusätzliche Schaltflächen für die PGP-Funktionen zur Verfügung (von links nach rechts):

Launch `PGPkeys` Application Startet `PGPkeys`, das Programm zur Schlüsselverwaltung. Näheres zu `PGPkeys` finden Sie in Kapitel 18.6 auf Seite 187.

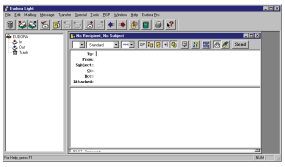


Abbildung 19.6: Eudora/ausgehende Mail

Use PGP/MIME When Sending Email Message Wenn diese Schaltfläche gedrückt ist, wird das PGP/MIME-Format für den Versand der E-Mail genutzt. Ob die Funktion standardmäßig eingestellt sein soll oder nicht, können Sie über die Grundeinstellungen ändern. Bitte lesen Sie hierzu im Abschnitt 21.3.1 auf Seite 248 nach.

PGP Encrypt Email Message Wenn diese Schaltfläche gedrückt ist, wird die ausgehende Mail automatisch verschlüsselt. Ob die Funktion standardmäßig eingestellt sein soll oder nicht, können Sie über die Grundeinstellungen ändern. Bitte lesen Sie hierzu im Abschnitt 21.3.3 auf Seite 249 nach.

Wenn die E-Mail-Adresse der Empfängerin im Benutzernamen eines Schlüssels in Ihrem Schlüsselbund vorkommt, so verschlüsselt PGP die E-Mail automatisch an diesen Schlüssel. Andernfalls erscheint das Auswahlfenster, in dem Sie einen Schlüssel auswählen können, an den verschlüsselt werden soll. Bitte lesen Sie hierzu Abschnitt 20.1 auf Seite 231.

PGP Sign Email Message Wenn diese Schaltfläche gedrückt ist, wird die ausgehende E-Mail automatisch mit Ihrem privaten Schlüssel signiert. Sie müssen hierzu allerdings noch das zu Ihrem Schlüssel passende Mantra eingeben (vgl. Abschnitt 20.2 auf Seite 234).

Wenn Sie in den Grundeinstellungen angegeben haben, daß Ihr Mantra zum Signieren für eine gewisse Zeit im Speicher gehalten werden soll und diese Zeit seit der letzten Signatur noch nicht abgelaufen ist, dann müssen Sie das Mantra nicht erneut eintippen. Beachten Sie hierzu bitte die Erläuterungen im Abschnitt 21.1.1.2 auf Seite 243.

19.4.2. PGP-Plugin für Microsoft Outlook/Exchange

Das PGP-Plugin für Microsofts E-Mailprogramme Microsoft Outlook/ Microsoft Exchange stellt innerhalb von Outlook bzw. Exchange die PGP-Funktionen zur Verfügung. Es kann in allen besprochenen PGP-Versionen über das Installationsprogramm von PGP mit installiert werden.

19.4.2.1. Hauptprogrammfenster

Im Menü des Programmfensters von Outlook/Exchange wird ein Punkt PGP eingefügt, der die folgenden Befehle enthält (Abb. 19.7):

`PGPkeys` **starten (Launch** `PGPkeys`**)** Startet das in Kapitel 18.6 auf Seite 187 beschriebene Programm zur Schlüsselverwaltung, `PGPkeys`.

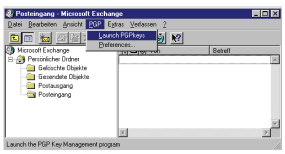


Abbildung 19.7: Menüpunkt PGP in Exchange

PGP Grundeinstellungen (Preferences) Startet das Fenster zum Ändern der PGP Grundeinstellungen. Hierzu lesen Sie bitte in Kapitel 21 auf Seite 241 nach. Insbesondere sei hier auf den Abschnitt 21.3 auf Seite 248 verwiesen, in dem das Verhalten des PGP-Plugins im Zusammenspiel mit dem E-Mailprogramm gesteuert wird.

Für den Start von `PGPkeys` (s. o.) steht Ihnen bei Microsoft Outlook im Haupt-Programmfenster zusätzlich zum oben beschriebenen Menübefehl auch eine Schaltfläche zur Verfügung.

19.4.2.2. Fenster für eingehende E-Mails

Im Fenster einer eingehenden E-Mail von Microsoft Outlook/Microsoft Exchange (Abb. 19.8) stehen Ihnen mit dem PGP-Plugin drei zusätzliche Schaltflächen zur Verfügung (von links nach rechts):

Decrypt Message and Verify Signature Wenn Sie diese Schaltfläche betätigen, wird der Dialog zur Entschlüsselung und/oder zur Signaturprüfung gestartet. Bitte lesen Sie hierzu in den Abschnitten 20.4 auf Seite 237 und 20.5 auf Seite 238 nach. Dieser Vorgang kann beim Öffnen einer Nachricht automatisch eingeleitet werden (nicht bei PGP 5.0i). Bitte beachten Sie hierzu Abschnitt 21.3 auf Seite 248.

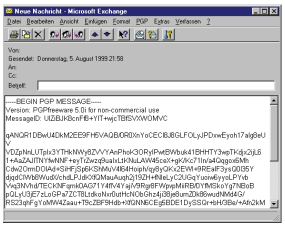


Abbildung 19.8: Exchange – Eingehende Nachrichten

Wenn Sie in den Grundeinstellungen angegeben haben, daß Ihr Mantra zum Entschlüsseln für eine gewisse Zeit im Speicher gehalten werden soll und diese Zeit seit der letzten Entschlüsselung noch nicht abgelaufen ist, dann müssen Sie das Mantra nicht nochmals eintippen. Beachten Sie hierzu bitte die Erläuterungen im Abschnitt 21.1.1.2 auf Seite 243.

Add Key to Keyring Wenn im E-Mail-Text ein PGP-Schlüssel enthalten ist, wird beim Klicken auf diese Schaltfläche der Schlüsselimport gestartet. Zum Schlüsselimport lesen Sie bitte im Abschnitt 18.2.3 auf Seite 178 nach. Dieser Vorgang kann beim Öffnen einer (unverschlüsselten, nicht signierten) Nachricht automatisch eingeleitet werden (nicht bei PGP 5.0i). Bitte beachten Sie hierzu Abschnitt 21.3 auf Seite 248.

Launch PGPkeys Startet das Programm zur Schlüsselverwaltung, `PGPkeys`, vgl. Kapitel 18.6 auf Seite 187.

Außerdem steht Ihnen im Fenster für eingehende E-Mails ein zusätzliches Menü unter dem Punkt PGP (Abb. 19.9) mit folgenden Befehlen zur Verfügung:

Decrypt/Verify Dieser Menübefehl ist gleichbedeutend mit dem oben beschriebenen Schaltflächen-Befehl `Decrypt Message and Verify Signature`.

Add Key Dieser Menübefehl ist gleichbedeutend mit dem oben beschriebenen Schaltflächen-Befehl `Add Key to Keyring`.

Launch `PGPkeys` Startet das Programm zur Schlüsselverwaltung, `PGPkeys`, beschrieben in Kapitel 18.6 auf Seite 187.

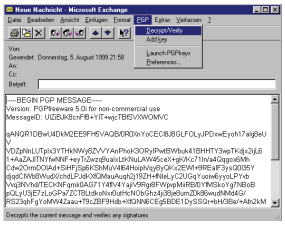


Abbildung 19.9: Exchange – Menüpunkt PGP bei eingehenden Mails

Preferences Startet das Fenster zum Ändern der PGP-Grundeinstellungen. Die Beschreibung finden Sie in Kapitel 21 auf Seite 241. Insbesondere sei hier auf den Abschnitt 21.3 auf Seite 248 verwiesen, in dem das Verhalten des PGP-Plugins im Zusammenspiel mit dem E-Mailprogramm gesteuert wird.

19.4.2.3. Fenster für ausgehende E-Mails

Im Fenster einer ausgehenden E-Mail von Microsoft Outlook/Microsoft Exchange (Abb. 19.10) stehen Ihnen mit dem PGP-Plugin drei zusätzliche Schaltflächen zur Verfügung (von links nach rechts):

Encrypt Message before sending (Nachricht vor dem Versenden verschlüsseln) Wenn diese Schaltfläche gedrückt ist, wird die ausgehende Mail automatisch vor Versand verschlüsselt. Ob die Funktion standardmäßig eingestellt sein soll oder nicht, können Sie über die Grundeinstellungen ändern. Bitte lesen Sie hierzu im Abschnitt 21.3.3 auf Seite 249 nach.

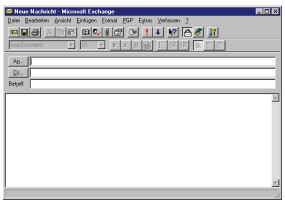


Abbildung 19.10: Exchange – ausgehende Mail

Wenn die E-Mail-Adresse der Empfängerin im Benutzernamen einer der Schlüssel in Ihrem Schlüsselbund vorkommt, so verschlüsselt PGP die E-Mail automatisch an diesen Schlüssel. Andernfalls erscheint das Auswahlfenster, in dem Sie einen Schlüssel auswählen können, an den verschlüsselt werden soll. Bitte lesen Sie hierzu Abschnitt 20.1 auf Seite 231.

Sign message before sending (Nachricht vor dem Versenden unterschreiben) Wenn diese Schaltfläche gedrückt ist, wird die ausgehende E-Mail automatisch vor dem Versand mit Ihrem privaten Schlüssel signiert. Sie müssen hierzu allerdings noch das zu Ihrem Schlüssel passende Mantra eingeben (siehe auch Abschnitt 20.2 auf Seite 234).

Wenn Sie in den Grundeinstellungen angegeben haben, daß Ihr Mantra zum Signieren für eine gewisse Zeit im Speicher gehalten werden soll und diese Zeit seit der letzten Signatur noch nicht abgelaufen ist, dann müssen Sie das Mantra nicht wieder eintippen. Beachten Sie dazu aber bitte die Erläuterungen im Abschnitt 21.1.1.2 auf Seite 243.

Launch `PGPkeys` (`PGPKeys` starten) Startet das Programm zur Schlüsselverwaltung, `PGPkeys`, dessen Bedienung in Kapitel 18.6 auf Seite 187 beschrieben ist.

Außerdem steht Ihnen im Fenster für ausgehende E-Mails ein zusätzliches Menü unter dem Punkt PGP (Abb. 19.11) mit folgenden Befehlen zur Verfügung:

Encrypt on send (Beim Versenden verschlüsseln) Dieser Menübefehl ist gleichbedeutend mit der oben beschriebenen Schaltfläche `Encrypt Message before sending`.

Sign on send (Beim Versenden unterschreiben) Dieser Menübefehl ist gleichbedeutend mit der oben beschriebenen Schaltfläche `Sign Message before sending`.

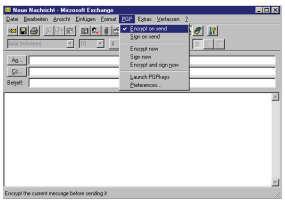


Abbildung 19.11: Exchange – Menü für ausgehende Nachrichten

Encrypt now (Jetzt verschlüsseln) Dieser Menübefehl leitet die Verschlüsselung sofort ein (nicht erst beim Versand der E-Mail). Wenn die E-Mail-Adresse der Empfängerin im Benutzernamen eines Schlüssels in Ihrem Schlüsselbund vorkommt, so verschlüsselt PGP die E-Mail automatisch an diesen Schlüssel. Andernfalls erscheint das Auswahlfenster, in dem Sie einen Schlüssel auswählen können, an den verschlüsselt werden soll. Bitte lesen Sie hierzu Abschnitt 20.1 auf Seite 231.

Sign now (Jetzt signieren) Dieser Menübefehl leitet das Unterschreiben sofort ein (nicht erst beim Versand der E-Mail). Sie müssen hierzu allerdings noch das zu Ihrem Schlüssel passende Mantra eingeben (siehe Abschnitt 20.2 auf Seite 234).

Wenn Sie in den Grundeinstellungen angegeben haben, daß Ihr Mantra zum Signieren für eine gewisse Zeit im Speicher gehalten werden soll und diese Zeit seit der letzten Signatur noch nicht abgelaufen ist, dann müssen Sie das Mantra nicht nochmals eintippen. Beachten Sie hierzu bitte die Erläuterungen und Überlegungen im Abschnitt 21.1.1.2 auf Seite 243.

Launch `PGPkeys` (`PGPkeys` starten) Startet `PGPkeys`, das Programm zur Schlüsselverwaltung. Näheres finden Sie in Kapitel 18.6 auf Seite 187.

Preferences (Grundeinstellungen) Startet das Fenster zum Ändern der PGP-Grundeinstellungen. Hierzu lesen Sie bitte in Kapitel 21 auf Seite 241. Insbesondere sei hier auf den Abschnitt 21.3 auf Seite 248 verwiesen, in dem das Verhalten des PGP-Plugins im Zusammenspiel mit dem E-Mailprogramm gesteuert wird.

19.4.3. PGP-Plugin für Pegasus Mail

`QDPGP` stellt die Funktionen von PGP in Pegasus Mail zur Verfügung. Für verschiedene Versionen von PGP und Pegasus gibt es unterschiedliche Plugins.

Die Version 3 von `QDPGP` benötigt mindestens Pegasus 3.0 und PGP 6.0i.

Die Version 2.12 von `QDPGP` benötigt mindestens Pegasus 3.0 und PGP 5.5.3i.

Wenn Sie ältere Versionen von Pegasus oder PGP 5.0i oder PGP 2.6.xi einsetzen, müssen Sie die `QDPGP`-Version 1.71 verwenden. Die Installation und Bedienung sind hier beispielhaft an `QDPGP` Version 2.12 erläutert. Alle Versionen sind auf der beiliegenden CD enthalten. Weitere Informationen und die Programmdateien finden Sie im Internet unter `http://community.wow.net`.

Die PGP-Plugins für die diversen Versionen von Pegasus Mail sind alle nicht in den PGP-Installationsprogrammen enthalten, Sie müssen allesamt getrennt installiert werden. Sie finden die PGP-Plugins für Pegasus Mail auf der beiliegenden CD.

19.4.3.1. Installation des Plugins

Das QDPGP-Plugin funktioniert nur mit der 32-Bit Version von Pegasus Mail, nicht für die 16-Bit Variante.

Voraussetzung für das Funktionieren des Plugins sind eine lauffähige Pegasus-Installation und eine lauffähige PGP-Installation in den benötigten Versionen.

Das Pegasus-PGP-Plugin kommt in Form einer Zip-komprimierten Datei, die Sie auch auf der beiliegenden CD finden (das entsprechende Verzeichnis heißt `Plugin/Win32/Pegasus/`). Diese Datei muß für die Installation zuerst mit einer entsprechenden Entpacker-Software, die Zip-Archive dekomprimieren kann (z. B. der Freeware `Alladin Expander`), in ein beliebiges (am besten leeres) Verzeichnis auf der Festplatte ausgepackt werden. Die Entpacker-Software muß mit den langen Dateinamen von Windows 95/98/NT umgehen können, um eine ordnungsgemäße Installation zu gewährleisten.

Die vom Entpacker-Programm erzeugten Dateien werden nur für die Installation benötigt und können nach erfolgreicher Installation des Plugins wieder gelöscht werden. Näheres zum Dekomprimieren des Zip-Archivs entnehmen Sie bitte der Dokumentation des jeweiligen von Ihnen benutzten Programmes.

Nach dem Auspacken finden sich im Zielverzeichnis folgende Dateien und Verzeichnisse:

`install.exe` Das Installationsprogramm, das die benötigten Programmdateien in komprimierter Form enthält. Es handelt sich um eine selbstextrahierende Datei, die nach dem Aufruf die Installation automatisch durchführt.

`install.sig` Eine abgetrennte PGP-Signatur zur Datei `install.exe`, mit der die Authentizität von `install.exe` überprüft werden kann (ein korrekt installiertes PGP vorausgesetzt – aber das ist ohnehin Voraussetzung für die Installation).

`readme.txt` Eine Textdatei, in der einige Informationen zum Programm, zur Installation und zu verschiedenen Versionen aufgeführt sind.

`file-id.diz` Eine Datei, in der eine kurze Information über das Programmpaket enthalten ist.

Durch einen Doppelklick auf das Symbol von `install.exe` starten Sie das Installationsprogramm. Als erstes erscheint eine Nachfrage, ob Sie das Programm wirklich installieren möchten. Hier bestätigen Sie mit `Ja`. Im nächsten Fenster wird Ihnen die Lizenzvereinbarung angezeigt, die Sie lesen und anschließend ebenfalls mit `Ja` bestätigen können. Danach beginnt das Installationsprogramm mit dem Kopieren der Dateien und den Eintragungen in der Windows-Registry-Datei. Wenn die Installation abgeschlossen ist, erscheint eine entsprechende Meldung, die Sie mit `OK` bestätigen. Danach ist das `QDPGP`-Plugin bereit für den Gebrauch.

19.4.3.2. Hauptprogrammfenster

Im Programmfenster von Pegasus Mail (Abb. 19.12) werden durch das PGP-Plugin zwei zusätzliche Schaltflächen eingefügt:

Launch `PGPkeys` Startet `PGPkeys`, das in Kapitel 18.6 auf Seite 187 beschriebene Programm zur Schlüsselverwaltung.

QDPGP Startet die Einstellungsmaske für die Einbindung der Verschlüsselung in Pegasus Mail. Diese Maske ist in Abschnitt 19.4.3.5 auf Seite 227 beschrieben.

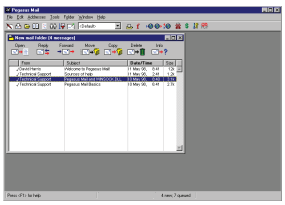


Abbildung 19.12: Pegasus/eingehende Mail

Außerdem wird in das Menü `Tools/Extensions` von Pegasus Mail der Menüpunkt `PGPkeys` eingetragen, über den Sie `PGPkeys` starten können.

19.4.3.3. Eingehende E-Mails

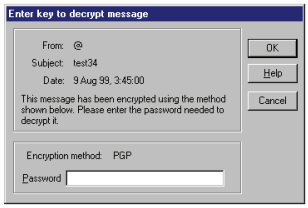


Abbildung 19.13: Pegasus – Mantra-Abfrage

Wenn eine eingegangene E-Mail verschlüsselt ist, dann zeigt Ihnen Pegasus ein Fenster, in dem Sie aufgefordert werden, das Mantra einzugeben (Abb. 19.13). Wenn Sie das zu dem benötigten Schlüssel gehörige Mantra richtig eingegeben haben, wird Ihnen die E-Mail im Klartext angezeigt. Wenn Sie das Mantra falsch eingegeben haben oder der benötigte private Schlüssel sich nicht in Ihrem Schlüsselbund befindet, dann erscheint eine Meldung, daß das Mantra falsch war und die Entschlüsselung fehlgeschlagen ist. Ihnen

wird dann in einem Fenster der verschlüsselte Text anstatt des Klartextes angezeigt. Leider wird nicht angezeigt, welcher private Schlüssel benötigt wird, im Zweifelsfall bleibt Ihnen daher nichts anderes übrig, als wiederholt „Entschlüsseln“ aufzurufen.

Gleichzeitig überprüft das Pegasus-Plugin verschlüsselte E-Mails auf Signaturen. Das PGP-Plugin zeigt das Ergebnis der Signaturprüfung in einem Dialogfenster an (Abb. 19.14). `Good signature` bedeutet, daß die Unterschrift zu den Daten paßt, diese also nicht verändert worden sind. `Message not signed` bedeutet, daß die Nachricht keine Signatur beinhaltet, die geprüft werden könnte.

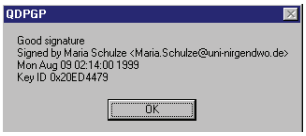


Abbildung 19.14: Pegasus – Ergebnis der Signaturprüfung

Wenn eine eingehende E-Mail nur signiert, aber nicht verschlüsselt ist, dann zeigt Pegasus die Nachricht samt Signatur an. Wenn Sie die Signatur prüfen möchten, müssen Sie dies über die PGP-Funktionen der Zwischenablage tun. Das PGP-Plugin für Pegasus prüft Nachrichten, die nur signiert sind, nicht automatisch. Für die Überprüfung von Signaturen über die Zwischenablage lesen Sie bitte Abschnitt 19.3.4 auf Seite 212 und Abschnitt 20.5 auf Seite 238.

19.4.3.4. Ausgehende E-Mails

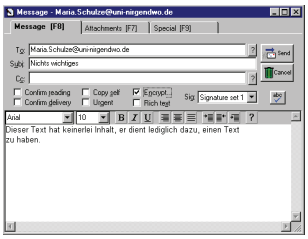


Abbildung 19.15: Ausgehende Nachricht (Pegasus)

Wenn Sie eine neue Nachricht erzeugen oder eine andere Nachricht weiterleiten, dann zeigt Ihnen Pegasus ein Fenster an, in dem Sie mehrere Angaben zu der ausgehenden Nachricht eingeben müssen (Abb. 19.15). Wenn Sie die Funktionen von PGP mit der Nachricht verwenden möchten, müssen Sie hier den Punkt `verschlüsseln` (`encrypt`) durch Mausklick in das entsprechende Feld aktivieren. Im Feld mit der Empfängeradresse muß für PGP-Verschlüsselung eine E-Mail-Adresse stehen, die im Benutzernamen eines Schlüssels in Ihrem Schlüsselbund vorkommt, da das Pegasus-Plugin darüber den benötigten Schlüssel auswählt. Eine Verschlüsselung an abweichende Schlüsselkennungen, die von Hand in die Liste eingetragen werden, ist beim Pegasus-Plugin nicht möglich. Findet PGP die E-Mail-Adresse des Empfängers nicht im

Schlüsselbund, dann bricht die Verschlüsselung der E-Mail beim Abschicken mit einer Fehlermeldung ab.

Wenn Sie das Feld `encrypt` anklicken, um die Verschlüsselung zu aktivieren, dann erscheint ein Dialogfenster mit Optionen zur Verschlüsselung (Abb. 19.16). Hier müssen Sie zuerst aus dem Listenfeld mit dem Titel `Encryption method` den Punkt `PGP` auswählen, wenn Sie PGP mit Pegasus benutzen möchten.

Abbildung 19.16: Verschlüsselungsoptionen (Pegasus)

Dann können Sie mit Hilfe der Optionsfelder `Encrypt message` (Nachricht verschlüsseln) und `Add digital signature` (Signatur hinzufügen) auswählen, ob Sie die Nachricht nur verschlüsseln oder nur signieren oder verschlüsseln und signieren möchten. Wenn Sie die Nachricht signieren wollen, müssen Sie im darüber liegenden Feld `Password` das Mantra zu Ihrem PGP-Schlüssel eingeben, der für die Signatur benutzt wird. Wenn Sie dies nicht tun oder das Mantra falsch eingeben, schlägt die Signierung beim Senden der E-Mail fehl und der Versand bricht mit einer Fehlermeldung ab.

Wenn Sie alle Eingaben gemacht haben, bestätigen Sie mit `OK`. Sie gelangen zurück zum Editierfenster der E-Mail, von dem aus Sie in das Verschlüsselungs-Optionsfenster gelangt sind. Hier können Sie nun Ihre E-Mail weiter bearbeiten, wenn Sie dies wünschen. Wenn Sie damit fertig sind und auf die Schaltfläche `Send` (Abschicken) klicken, startet das Plugin die Verschlüsselung und/oder das Unterschreiben der Nachricht. Sollte dabei ein Fehler auftreten, wird der Vorgang abgebrochen.

19.4.3.5. Einstellungen QDPGP

Wenn Sie über die entsprechende Schaltfläche im Hauptfenster von Pegasus Mail das Programm `QDPGP` aufrufen, erhalten Sie ein Fenster mit einigen Einstellungen (Abb. 19.17).

Mit den Schaltflächen an der linken Seite können Sie, von oben nach unten, die folgenden Aktionen durchführen:

Hilfe/Help Soll die Online-Hilfe zum PGP-Plugin für Pegasus Mail anzeigen; die Schaltfläche ist jedoch in den von uns getesteten Versionen funktionslos.

Version Versucht, eine Internetverbindung zur URL `http://community.wow.net` aufzubauen, um dort nach einer aktuellen Version des Plugins zu suchen. Bevor das geschieht, erfolgt noch eine Rückfrage.

Abbildung 19.17: Grundeinstellungen QDPGP

Eingabe/Enter Beendet das `QDPGP`-Fenster und übernimmt evtl. geänderte Einstellungen.

Abbrechen/Cancel Beendet das `QDPGP`-Fenster und verwirft die vorgenommenen Änderungen.

Folgende Einstellungen können vorgenommen werden:

Verwende PGP-Schlüssel default/Use `PGPkeys` default Wenn diese Option aktiviert ist, dann benutzt das Pegasus Plugin den in `PGPkeys` als Standardschlüssel gesetzten Schlüssel ebenfalls als Standardschlüssel für die Signatur von E-Mails. Wenn diese Option aktiv ist, dann ist automatisch das darunterliegende Eingabefeld für die als Standard zu benutzende Schlüsselnummer oder den Benutzernamen deaktiviert. Dasselbe gilt für die Option `Aktuelle Identität/Current Identity` im Bereich `Optionen`, da dann der in `PGPkeys` eingestellte Standardschlüssel benutzt wird.

Benutzernamen oder Schlüssel-Nummer Wenn die oben genannte Option `Verwende PGP-Schlüssel default` nicht aktiviert ist, kann in diesem Feld ein Benutzername oder eine eindeutige Schlüsselnummer eingegeben werden. Der dazugehörige Schlüssel wird dann innerhalb von Pegasus als Standard-Schlüssel benutzt. Auf die Standardeinstellungen von PGP zur Benutzung außerhalb von Pegasus Mail hat dies keine Auswirkungen.

Kommentar/Comment Hier kann ein Kommentar eingegeben werden, der bei der Verschlüsselung in die verschlüsselten Daten eingefügt wird. Entspricht der in Abschnitt 21.1.1.5 auf Seite 244 beschriebenen Funktion.

Aktuelle Identität/Current Identity Wenn diese Option aktiviert ist, wird der Standardschlüssel für die Signatur von Daten mit dem PGP-Plugin für Pegasus Mail in Abhängigkeit von der bei der Anmeldung angegebenen Identität festgelegt (für Mehrbenutzerinstallationen von Pegasus). So benutzt jede Benutzerin immer ihren eigenen Schlüssel, die Identität wird bei der Pegasus-Anmeldung geklärt. Diese Option kann nur gewählt werden, wenn die Option `Verwende PGP-Schlüssel Default` ausgeschaltet ist.

VerschlüsseleDef/Encrypt Default Entspricht der in Abschnitt 21.1.1.1 auf Seite 241 beschriebenen Option. Wenn diese Option aktiviert ist, wird immer auch an den eigenen Schlüssel mitverschlüsselt.

Vorsicht bei der Mischung von RSA- und DSS/ElGamal-Schlüsseln! Wenn Sie einen ElGamal-Schlüssel verwenden, führt die Verwendung dieser Option leider dazu, daß Empfängerinnen mit PGP 2.6.x Ihre Nachrichten nicht entschlüsseln können. Näheres finden Sie auf Seite 241.

PGP Version 5.5/6.0/7.0 Hier zeigt das Plugin an, welche PGP-Version es benutzt. Da Sie immer nur eine Version von PGP für Windows installiert haben können, können Sie diese Einstellung nicht ändern. Ob `QDPGP` tatsächlich mit PGP 7.0 wird arbeiten können, bleibt abzuwarten, noch existiert diese PGP-Version nicht.

Debug Wenn diese Option aktiviert ist, zeigt das Plugin bei Fehlern ausführlichere Meldungen an, um Sie bei der Fehlerbehebung zu unterstützen.

Web-Hilfe/Web-Help Wenn diese Option aktiviert ist, versucht das Plugin, nicht die Online-Hilfe auf dem lokalen Rechner zu benutzen, sondern stattdessen die Hilfedaten aus dem Internet anzuzeigen (vom Server `community.wow.net`).

Rüstung/Armor 7-bit-Kodierung verwenden, damit die Daten beim Transport per E-Mail nicht beschädigt werden. Entspricht der Option `Text Output beim Verschlüsseln von Dateien` mit den Explorer-Erweiterungen von PGP. Näheres finden Sie im Abschnitt 13.10 auf Seite 93.

Abtrennen/Detach Wenn diese Option gewählt ist, soll das Plugin abgetrennte Signaturdateien erzeugen. Das ist für E-Mails im allgemeinen nicht sinnvoll, diese Option ist daher nicht anwählbar.

Wählen/Select Es ist uns nicht bekannt, was diese Option bewirken soll; sie ist anscheinend funktionslos.

Sprache/Language Hier können Sie über das Listenfeld aus mehreren Sprachversionen für das PGP-Plugin auswählen. Leider betrifft die Sprache nur das in diesem Abschnitt beschriebene Einstellungsfenster, nicht die übrigen Meldungen in Pegasus Mail, die von dieser Einstellung unberührt bleiben.

20. PGP benutzen – Aktionen durchführen

20.1. Daten verschlüsseln

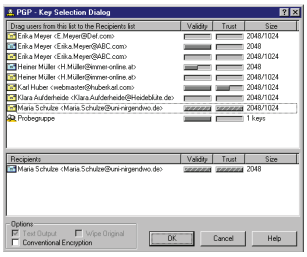


Abbildung 20.1: Auswahl der Empfänger (5.5.3i)

Sie haben auf einem der Wege, die in den vorangegangenen Abschnitten beschrieben worden sind, die Verschlüsselung von Daten (entweder aus der Windows-Zwischenablage oder von Dateien) veranlaßt. PGP zeigt Ihnen nun ein Fenster an, in dem Sie festlegen können, an wen es die Daten verschlüsseln soll (Abb. 20.1).

Wenn Sie E-Mails aus einem Programm mit installiertem PGP-Plugin verschicken und die E-Mail-Adresse des Empfängers in den Benutzerkennungen der in Ihrem Schlüsselbund vorhandenen Schlüssel enthalten ist, dann entfällt dieser Schritt, da PGP den benötigten Schlüssel über die E-Mail-Adresse herausfinden kann. Wenn jedoch kein passender Schlüssel vorhanden ist, erscheint auch in diesem Fall das Schlüssel-Auswahl-Fenster. Beim Verschlüsseln von Dateien und Daten der Zwischenablage über `PGPtray`, die PGP Explorer-Erweiterungen oder `PGPtools` müssen Sie den Schlüssel immer manuell auswählen.

Die Schlüssel, an die die Daten verschlüsselt werden sollen, wählen Sie aus, indem Sie den jeweiligen Schlüssel in der Anzeigeliste in der oberen Hälfte des Fensters mit der Maus markieren und dann den Schlüssel mit gezogener Maustaste auf das Feld mit der Empfängerliste (`Recipient List`) ziehen und die Maustaste dort loslassen („drag & drop“). Sie können auf diese Weise mehrere Schlüssel auswählen, an die verschlüsselt werden soll. Dann kann jede dieser Benutzerinnen die Daten mit ihrem privaten Schlüssel wieder entschlüsseln, solange Sie nicht unterschiedliche Schlüsselarten gemischt haben (siehe die nachfolgenden Absätze).

Zusätzlich zu den Schlüsseln können Sie im Schlüssel-Auswahl-Fenster noch folgende Optionen angeben:

Textausgabe (Text Output) Wenn Sie eine Datei über `PGPtools` oder die Explorer-Erweiterungen verschlüsseln, können Sie PGP mit dieser Option dazu bringen, die verschlüsselten Daten im 7-Bit-ASCII-Format auszugeben. Hiermit können Sie ganz sicher sein, daß die verschlüsselte Datei einen Transport per E-Mail unbeschadet übersteht, dafür wächst die Dateigröße der verschlüsselten Datei um etwa 30% an.

Bei Verschlüsselung von Daten aus der Zwischenablage oder per Plugin in Ihrem E-Mailprogramm ist diese Option immer aktiviert und kann nicht abgeschaltet werden (bei PGP 6.0i wird Sie dann gar nicht erst angezeigt). Wenn Sie die verschlüsselte Datei per E-Mail verschicken möchten, sollten Sie diese Option aktivieren. Näheres zu diesem Ausgabeformat finden Sie in Abschnitt 13.10 auf Seite 93.

Original löschen (Wipe Original) Wenn Sie eine Datei verschlüsseln, können Sie mit dieser Option PGP dazu bringen, das unverschlüsselte Original nach der Verschlüsselung automatisch zu löschen. Gehen Sie mit dieser Option vorsichtig um, um ungewollten Datenverlust zu vermeiden!

Bei PGP 5.0i steht diese Funktion nicht zur Verfügung, bei PGP 6.0i wird sie nur angezeigt, wenn eine Datei verschlüsselt wird, nicht beim Verschlüsseln bspw. der Zwischenablage.

Konventionelle Verschlüsselung (Conventional Encryption) (nicht bei PGP 5.0i) In den PGP-Versionen ab 5.5.3 können Sie beim Verschlüsseln auch (wie schon bei 2.x) auswählen, daß die Verschlüsselung nicht über das Verfahren mit öffentlichem und geheimen Schlüsselbund erfolgen soll, sondern über ein Mantra, das sowohl die Absenderin als auch der Empfänger kennen müssen, es muß also auf sicherem Wege übertragen werden.

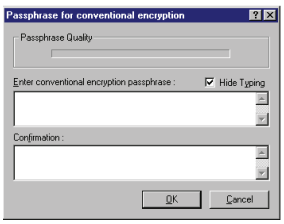


Abbildung 20.2: Konventionelle Verschlüsselung (5.5.3i)

Diese Option ist für E-Mail nicht empfehlenswert, da Sie den großen Vorteil der asymmetrischen Verschlüsselung außer Funkti-

on setzt, nämlich daß der Schlüssel selbst nicht auf sicherem Wege übertragen werden muß. Für das Verschlüsseln von Dateien zur Aufbewahrung auf dem eigenen Rechner hingegen ist die Option sehr sinnvoll.

Wenn Sie sich für die Aktivierung der Konventionellen Verschlüsselung entscheiden, dann werden die Schlüssellisten mit zur Verfügung stehenden Schlüsseln und Empfängern ausgeblendet. Nach dem Bestätigen mit `OK` erscheint ein Fenster, in dem Sie das Mantra für die konventionelle Verschlüsselung zweimal eingeben müssen (Abb. 20.2). Zu der Anzeige „passphrase quality“ gilt das auf Seite 170 gesagte. In PGP 5.0i können Sie zwar keine Daten mit dieser Option verschlüsseln, Sie können derart verschlüsselte Daten aber wieder entschlüsseln, wenn Ihnen das jeweilige Mantra bekannt ist.

Wenn alle gewünschten Empfänger in der Empfängerliste enthalten sind, können Sie die Verschlüsselung mit einem Klick auf die Schaltfläche `OK` starten. Falls Sie in der Empfängerliste RSA- und DSS/ElGamal-Schlüssel gemischt haben sollten, zeigt Ihnen PGP eine Meldung an, daß Benutzer von PGP-Versionen vor 5.0 die Daten nicht entschlüsseln können. Da dies leider den Tatsachen entspricht, sollten Sie nur dann verschiedene Schlüsselarten mischen, wenn Sie ganz sicher sind, daß alle Empfängerinnen PGP-Versionen ab 5.0 oder andere OpenPGP-kompatible Programme wie z. B. GnuPG einsetzen. Davon können Sie zur Zeit jedoch nicht ausgehen, es benutzen immer noch sehr viele Leute (teilweise aus guten Gründen) PGP 2.6.xi. Sie sollten also, wenn Sie Daten an Benutzer mit RSA-Schlüsseln und an Benutzerinnen mit DSS/ElGamal-Schlüsseln verschlüsseln möchten, dies in zwei getrennten Verschlüsselungsvorgängen für die beiden Schlüsselarten durchführen, dann gibt es keine Probleme.

Die eigentliche Verschlüsselung ist damit abgeschlossen. Wenn Sie die Funktion für Daten in der Windows-Zwischenablage aufgerufen haben, stehen die verschlüsselten Daten jetzt anstelle der Originaldaten in der Zwischenablage und Sie können Sie in Ihrem Anwendungsprogramm (z. B. E-Mail- oder Textverarbeitungsprogramm) mit den normalen Windows-Funktionen `Einfügen` bzw. `Paste` einsetzen.

Falls Sie eine Datei verschlüsselt haben, wird das Ergebnis der Verschlüsselung unter dem Namen der Ursprungsdatei mit angehängtem `.pgp` im selben Verzeichnis wie die Ursprungsdatei abgespeichert. Aus

einer Datei `Text.txt` wird also die Datei `TEXT.TXT.pgp`. Wenn der betreffende Dateiname im jeweiligen Verzeichnis bereits vorhanden ist, erscheint ein `Speichern unter`-Fenster, in dem Sie einen anderen Namen angeben, ein anderes Verzeichnis auswählen oder, wenn Sie den Namen beibehalten möchten, die bereits bestehende Datei diesen Namens überschreiben können.

Bei PGP 5.0i erscheint dieses `Speichern unter`-Fenster immer, auch wenn der betreffende Dateiname noch nicht vorhanden ist.

20.2. Daten signieren

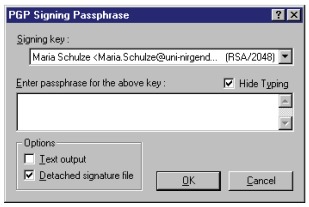


Abbildung 20.3: Daten signieren (5.5.3i)

Wenn Sie auf einem der in den vorangegangenen Abschnitten beschriebenen Wege das Unterschreiben einer Datei, der Zwischenablage oder einer E-Mail veranlaßt haben, erscheint ein Fenster, in dem Sie festlegen können, mit welchem privaten Schlüssel die Daten signiert werden sollen (Abb. 20.3).

Wenn Sie mehrere private Schlüssel in Ihrem Schlüsselbund haben, können Sie den für die Signatur gewünschten über das Listenfeld mit der Überschrift `Signing Key` auswählen. Um die Daten mit dem jeweils gewählten privaten Schlüssel zu signieren, müssen Sie danach im darunterliegenden Feld mit der Überschrift `Enter passphrase for the above key` das zum ausgewählten Schlüssel passende Mantra eingeben. Wenn die Option `Hide Typing` in diesem Fenster aktiviert ist (dies ist die Standardeinstellung), dann wird der Text Ihres Mantras nicht auf dem Bildschirm angezeigt, damit niemand das Mantra ablesen kann, der sich in der Nähe befindet.

Scheuen Sie sich nicht, Anwesende darum zu bitten, sich kurz umzudrehen, das ist auch beim Chef oder guten Freunden nicht unhöflich – aufmerksame, höfliche Menschen drehen sich ohnehin von selbst kurz weg, wenn sie bemerken, daß Sie irgendeine Art von Paßwort eingeben.

Wenn Sie eine Datei nur signieren, aber nicht verschlüsseln möchten, stehen Ihnen in diesem Fenster noch die Optionen Textausgabe (`Text Output`) und Abgetrennte Signaturdatei (`Detached Signature File`) zur Verfügung.

Die Bedeutung der Option *Textausgabe* wurde im vorangegangenen Abschnitt 20.1 auf Seite 231 erläutert.

Die Option *Abgetrennte Signaturdatei* erzeugt, wenn Sie aktiviert ist, eine von der Ursprungsdatei getrennte Signaturdatei. Die Ursprungsdatei bleibt dabei unverändert, auch eine Binärdatei kann von den Empfängerinnen verwendet werden, selbst wenn diese PGP nicht verwenden. Eine Prüfung der Signatur ist ohne PGP natürlich nicht möglich.

Wird keine abgetrennte Signaturdatei erzeugt, sondern die Signatur in die Ursprungsdatei eingefügt, so können Binärdaten (wie z. B. Bilddaten oder ausführbare Programme) nur von Leuten benutzt werden, die ebenfalls PGP verwenden, denn nur sie sind in der Lage, die Signatur von den eigentlichen Binärdaten zu trennen. Diese Option ist extrem sinnvoll, wenn Sie z. B. selbstgeschriebene Programme allgemein zur Verfügung stellen wollen – vermutlich haben Sie auch schon auf einem ftp-Server Dateien mit der Endung `.asc` oder `.sig` gesehen, die zusätzlich zu den eigentlichen Archiven zur Verfügung standen. Mit diesen abgetrennten Unterschriften können Sie sicherstellen, daß das Archiv, welches Sie vom Server geladen haben, tatsächlich vom Programmautoren stammt.

Wenn Sie in einem Arbeitsgang eine Datei sowohl verschlüsseln als auch signieren, stehen Ihnen diese beiden Optionen nicht zur Verfügung. Hier wählen Sie bereits im Verschlüsselungsdialog aus, ob die Ausgabedatei im 7-Bit Format abgespeichert werden soll. Eine abgetrennte Signatur ist in diesem Fall nicht vorgesehen. Dies hat insofern keine Bedeutung, als die verschlüsselte Datei sowieso nur von der Inhaberin des passenden privaten Schlüssels wieder gelesen werden kann. Da diese Person zum Entschlüsseln ohnehin PGP verwenden muß, ist sie automatisch auch in der Lage, die Signatur von den Daten zu trennen. Außerdem wird bei PGP die Unterschrift zusammen mit den Daten verschlüsselt. Mit `OK` schließen Sie die Signatur der Daten ab, mit der Schaltfläche `Cancel` brechen Sie ab, ohne eine Signatur zu erzeugen.

Wenn Sie Text signieren, dann läßt PGP die Ursprungsdaten im Klartext$^\vee$ und hängt daran die Signatur (im 7-Bit Format für E-Mail-Versand) an. Auf diese Weise kann die entsprechende Nachricht auch von Personen gelesen werden, die nicht mit PGP arbeiten, diese können allerdings die Signatur nicht überprüfen. Wenn Sie Daten in der Zwischenablage verschlüsseln und signieren, erfolgt die Ausgabe ebenfalls mit der Option Textausgabe – natürlich ohne lesbaren Klartext.

$\vee$ PGP 5.0i hat hier einen kleinen Fehler – Texte mit Umlauten werden nicht als Texte erkannt.

Wenn Sie eine Datei verschlüsselt und signiert haben, wird das Ergebnis der Verschlüsselung unter dem Namen der Ursprungsdatei (in Großbuchstaben) mit angehängtem `.pgp` im selben Verzeichnis wie die Ursprungsdatei abgespeichert, genau wie bei Verschlüsselung ohne Signatur. Aus einer Datei `Text.txt` wird also die Datei `TEXT.TXT.pgp`. Wenn der betreffende Dateiname im jeweiligen Verzeichnis bereits vorhanden ist, erscheint ein `Speichern unter`-Fenster, in dem Sie einen anderen Namen angeben, ein anderes Verzeichnis auswählen oder, wenn Sie den Namen beibehalten möchten, die bereits bestehende Datei diesen Namens überschreiben können.

Bei PGP 5.0i erscheint dieses `Speichern unter`-Fenster immer, auch wenn der betreffende Dateiname noch nicht vorhanden ist.

Falls Sie eine abgetrennte Signaturdatei erzeugt haben, wird diese unter dem Namen der signierten Datei⊕ mit angehängtem `.sig` im selben Verzeichnis wie die signierte Datei abgespeichert. Zu einer Datei `Text.txt` wird also die Signaturdatei `TEXT.TXT.sig` erzeugt. Auch hier gilt das eben gesagte über das `Speichern unter`-Fenster.

Sie können sowohl die Signaturdateien als auch die signierten Dateien umbenennen, ohne die Gültigkeit der Signatur zu beeinträchtigen. Allerdings geht dadurch die automatische Zuordnung von Signatur zu signierter Datei verloren, Sie müssen dann bei der Signaturprüfung angeben, zu welcher Datei die Signatur gehört. Wenn Sie Signaturdateien für Benutzer erzeugen möchten, die mit MS-DOS oder Windows 3.x arbeiten, das keine langen Dateinamen beherrscht, sollten Sie das dennoch tun, da sich diese Empfänger sonst mit den doch recht kryptischen MS-DOS-Abkürzungen der langen Dateinamen herumschlagen müssen. Aus `Erikas Text.txt.sig` wird unter MS-DOS so etwas wie `erikas~1.sig`. Wenn Sie diesen Empfängern eine automatische Zuordnung vereinfachen wollen, nehmen Sie als Dateinamen `erikatxt.txt` und `erikatxt.sig`.

⊕ Wenn der Dateiname der Ursprungsdatei der „8+3"-Konvention von Dateinamen unter MS-DOS und Windows 3.x entspricht, wird er dabei scheinbar in Großbuchstaben umgewandelt. Solche Dateinamen enthalten nur Großbuchstaben, werden aber vom Windows-Explorer so angezeigt, daß der erste Buchstabe groß ist und die folgenden Buchstaben klein. Wird nun die Endung `.pgp` hinzugefügt, so wird ein langer Dateiname daraus (wegen des zweiten Punktes und evtl. der Länge). PGP zeigt den ursprünglichen Dateinamen innerhalb des neuen Dateinamens in den Großbuchstaben an, aus denen er in Wirklichkeit besteht, daher die scheinbare Umwandlung.

20.3. Daten verschlüsseln und signieren

Haben Sie auf einem der oben genannten Wege die Verschlüsselung und gleichzeitige Signierung von Daten (entweder aus der Windows-Zwischenablage oder von Dateien) veranlaßt, startet PGP automatisch nacheinander zuerst den Dialog für die Verschlüsselung und anschließend den Dialog für die Signatur. Bitte lesen Sie hierzu die Abschnitte 20.1 auf Seite 231 und 20.2 auf Seite 234. Die Reihenfolge der Dialoge ist technisch gesehen verdreht, da PGP Daten zuerst signiert und anschließend verschlüsselt, so daß nur der vorgesehene Empfänger anhand der Unterschrift erkennen kann, wer die Nachricht versandt hat.

20.4. Verschlüsselte Daten wieder entschlüsseln

Abbildung 20.4: Entschlüsselungsmantra eingeben (5.5.3i)

Wenn Sie die Entschlüsselung von Daten aufrufen (siehe hierzu die Abschnitte 19.2.5 auf Seite 209, 19.3.4 auf Seite 212 und 19.1 auf Seite 206) und die Daten an einen Schlüssel verschlüsselt worden sind, der sich in Ihrem privaten Schlüsselbund befindet, so fordert PGP Sie auf, in einem Fenster das zum jeweiligen privaten Schlüssel passende Mantra einzugeben (Abb. 20.4). Wie in allen Fenstern von PGP, in denen Sie das Mantra Ihres privaten Schlüssels eingeben müssen, finden Sie auch hier die Option, die Anzeige des Mantras auf dem Bildschirm zu unterdrücken (Standardeinstellung). Im oberen Feld des Fensters werden Ihnen die Schlüssel angezeigt, an die die Daten verschlüsselt worden sind. Im unteren Feld des Fensters geben Sie das passende Mantra ein und bestätigen mit Klick auf die Schaltfläche `OK`.

Bei PGP 5.0i entfällt die Auflistung der Schlüssel. Wenn Sie unter PGP 5.0i mehrere private Schlüssel benutzen und nicht wissen, an welchen die betreffenden Daten verschlüsselt worden sind, müssen Sie leider ausprobieren, welches Mantra PGP akzeptiert (das heißt, welcher Schlüssel benutzt wurde).

Wenn Daten an mehrere Ihrer privaten Schlüssel verschlüsselt worden sind (beispielsweise an Ihren Privat- und Ihren Firmenschlüssel), so müssen Sie ein Mantra eingeben, das zu einem beliebigen der im oberen

Feld aufgelisteten Schlüssel paßt (natürlich nicht zu einem öffentlichen Schlüssel anderer Personen, die ebenfalls aufgelistet werden; aber deren Mantra kennen Sie ja auch nicht). PGP findet automatisch heraus, zu welchem Schlüssel das Mantra gehört und startet den Entschlüsselungsvorgang.

Wenn das Mantra richtig eingegeben wurde, wird die Entschlüsselung abgeschlossen, sonst erscheint eine entsprechende Fehlermeldung. Wenn Sie diese bestätigen, kommen Sie wieder zurück zum Eingabefenster und können das Mantra erneut eingeben.

Abbildung 20.5: Nachricht ist nicht für Sie verschlüsselt (5.5.3i)

Wenn sich kein passender privater Schlüssel in Ihrem Schlüsselbund befindet, so zeigt Ihnen PGP eine entsprechende Meldung an, daß Sie die Daten nicht entschlüsseln können (Abb. 20.5). In diesem Fall können Sie diese Meldung nur mit `Cancel` bestätigen und damit den Vorgang abbrechen. Wenn die Daten für Sie bestimmt waren, müssen Sie den Absender bitten, sie noch einmal mit Ihrem Schlüssel verschlüsselt zu schicken.

Falls Sie eine verschlüsselte Datei entschlüsselt haben, wird das Ergebnis der Entschlüsselung unter dem Namen der Ursprungsdatei im selben Verzeichnis wie die Ursprungsdatei abgespeichert. Die beim Verschlüsseln hinzugefügte Endung `.pgp` wird dabei wieder entfernt, der ursprüngliche Dateiname wird also wieder hergestellt, wenn die Datei nicht zwischenzeitlich umbenannt worden ist. Wenn der betreffende Dateiname im jeweiligen Verzeichnis bereits vorhanden ist, erscheint ein `Speichern unter`-Fenster, in dem Sie einen anderen Namen angeben, ein anderes Verzeichnis auswählen oder, wenn Sie den Namen beibehalten möchten, die bereits bestehende Datei dieses Namens überschreiben können.

Bei PGP 5.0i erscheint dieses `Speichern unter`-Fenster immer, auch wenn der betreffende Dateiname noch nicht vorhanden ist.

20.5. Signaturen prüfen

Die Prüfung von Signaturen (Aufruf siehe Abschnitte 19.1 auf Seite 206, 19.2.5 auf Seite 209, 19.3.4 auf Seite 212) nimmt PGP, wenn die Daten mit einem Schlüssel signiert worden sind, der sich in Ihrem Schlüsselbund

befindet, automatisch vor und zeigt Ihnen das Ergebnis der Signaturprüfung an.

Wenn es sich um eine abgetrennte Signaturdatei handelt, müssen Sie unter Umständen angeben, auf welche Datei sich die Signatur bezieht. Bei PGP 5.0i ist dies immer der Fall, bei den anderen Versionen nur dann, wenn sich keine Datei mit passendem Namen im Verzeichnis befindet, wo sich auch die Signaturdatei befindet. Wenn sich in diesem Verzeichnis eine Datei befindet, die denselben Namen wie die Signaturdatei hat, jedoch ohne `.sig` am Ende, nimmt PGP an, daß die Signaturdatei zu dieser Datei gehört. Wenn sich hier keine entsprechende Datei befindet, müssen Sie in dem erscheinenden Dateiauswahl-Fenster die Datei von Hand auswählen, zu der die Signatur gehört.

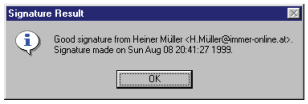


Abbildung 20.6: Signatur ist korrekt (5.0i)

PGP 5.0i zeigt Ihnen ein Dialogfenster mit dem Ergebnis der Prüfung in Textform (Abb. 20.6). Wenn die Signatur zu den Daten paßt, erscheint der Text `Good Signature from` (gültige Unterschrift von), gefolgt vom Benutzernamen der Unterschreibenden und Datum und Uhrzeit der Erzeugung. Wenn die Daten nicht zur Signatur passen, also nach dem Signieren verändert worden sind, erscheint der Text `Bad Signature from` (ungültige Unterschrift von), gefolgt vom Benutzernamen des Unterschreibenden und Datum und Uhrzeit der Erzeugung. Falls der öffentliche Teil des Schlüssels, mit dem die Signatur erzeugt wurde, sich nicht in Ihrem Schlüsselbund befindet, erscheint die Meldung `Could not find a public key to verify the signature on this message` (Konnte keinen öffentlichen Schlüssel finden, mit dem sich die Unterschrift prüfen läßt).

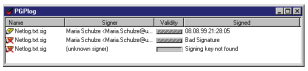


Abbildung 20.7: `PGPlog` – Ergebnis der Signaturprüfung (5.5.3i)

Bei PGP 5.5.3i und PGP 6.0i wird das Programm `PGPlog` gestartet, wenn Daten eine PGP-Signatur enthalten. In dessen Fenster (Abb. 20.7) wird das Ergebnis der Signaturprüfung angezeigt. Ein Bleistiftsymbol vor dem Dateinamen bzw. der Ursprungsbezeichnung (z. B. Clipboard für die Windows-Zwischenablage) in Kombination mit Angabe von Datum und Uhrzeit in der Spalte `Signed` (Signiert) bedeutet, daß die Daten zur Signatur passen. Wenn das Bleistiftsymbol durchgestrichen ist, konnte die Signatur entweder nicht überprüft werden, weil der dazu nötige Schlüssel nicht in Ihrem Schlüsselbund enthalten ist (dann steht in der Spalte `Signed` der Text `Signing`

key not found), oder die Signatur paßt nicht zu den Daten, weil diese nach der Erzeugung der Signatur verändert worden sind (dann steht in der Spalte Signed der Text Bad Signature).

Außer den obenstehenden Informationen über das Ergebnis der Signaturprüfung zeigt Ihnen PGPlog noch den Standard-Benutzernamen und die Gültigkeit des Schlüssels an, mit dem die Signatur erzeugt worden ist. Wenn der Schlüssel nicht in Ihrem Schlüsselbund enthalten ist, erscheint in der Spalte Benutzernamen in Klammern der Text unkown signer (unbekannter Unterzeichner).

Falls Sie die Signatur einer Datei geprüft haben, bei der die Signatur in der Datei enthalten ist (also keine abgetrennte Signatur), dann wird die Signatur aus der Datei bei der Prüfung entfernt und eine Ausgabedatei erzeugt, die die Daten ohne Signatur enthält. Diese Datei wird unter dem Namen der Ursprungsdatei im selben Verzeichnis wie die Ursprungsdatei abgespeichert. Die beim Signieren hinzugefügte Endung .pgp wird dabei wieder entfernt, der ursprüngliche Dateiname wird also wieder hergestellt, wenn die Datei nicht zwischenzeitlich umbenannt worden ist. Wenn der betreffende Dateiname im jeweiligen Verzeichnis bereits vorhanden ist, erscheint ein Speichern unter-Fenster, in dem Sie einen anderen Namen angeben, ein anderes Verzeichnis auswählen oder, wenn Sie den Namen beibehalten möchten, die bereits bestehende Datei diesen Namens überschreiben können.

Bei PGP 5.0i erscheint dieses Speichern unter-Fenster immer, auch wenn der betreffende Dateiname noch nicht vorhanden ist.

20.6. Entschlüsseln und Signatur prüfen

Rufen Sie auf einem der oben beschriebenen Wege diese Funktion auf, so startet PGP nacheinander die Dialoge zum Entschlüsseln und – wenn die Entschlüsselung erfolgreich verlaufen ist – zur Unterschriftenprüfung. Näheres zu diesen Dialogen finden Sie in den Abschnitten 20.4 auf Seite 237 und 20.5 auf Seite 238.

21. PGP-Grundeinstellungen

Über die Grundeinstellungen von PGP haben Sie bereits im Abschnitt 17.1 auf Seite 146 schon einmal zu lesen bekommen (Stichwort „Schnelle Schlüsselerzeugung durch vorausberechnete Primzahlen“). In diesem Kapitel sollen die übrigen Grundeinstellungen in Kurzform erläutert werden.

Sie kommen in das Fenster mit Grundeinstellungen, indem Sie entweder im Menü von `PGPtray` den Punkt `PGP Preferences` auswählen, oder indem Sie in `PGPkeys` den Menüpunkt `Edit/Preferences` auswählen.

Im Fenster `PGP Preferences` haben Sie mehrere Registerkarten zur Auswahl, der Inhalt ist je nach Version etwas unterschiedlich. Wenn eine Option nicht in allen Versionen zur Verfügung steht, ist das im Text entsprechend angegeben.

Innerhalb der Grundeinstellungen steht Ihnen für die einzelnen Punkte mit der rechten Maustaste eine kurze Erklärung der jeweiligen Einstellung zur Verfügung, oder über die Schaltfläche `Hilfe` eine etwas ausführlichere (aber leider nicht immer richtige) Erläuterung.

21.1. Registerkarte Allgemeines (General)

21.1.1. Verschlüsselungs- und Signatur-Grundeinstellungen (Encryption and Signing Preferences)

21.1.1.1. Immer an Standardschlüssel verschlüsseln (Always encrypt to default Key)

Standardmäßig ist diese Option ausgeschaltet.

Diese Option bedeutet, daß jede Verschlüsselung nicht nur an die ausgewählten Empfänger durchgeführt wird, sondern daß immer auch mit dem als Standardschlüssel gesetzten Schlüssel (also Ihrem eigenen Schlüssel) verschlüsselt wird. Dies bedeutet, daß die Daten bei Bedarf auch von Ihnen wieder entschlüsselt werden können. Diese Option

entspricht dem Eintrag `EncryptToSelf=on` in der Datei `config.txt` bei PGP 2.6.xi, siehe Abschnitt 14 auf Seite 106.

Anmerkung:
Wenn Daten sowohl mit DSS/ElGamal-Schlüsseln als auch mit RSA-Schlüsseln verschlüsselt werden, so können die dabei entstehenden verschlüsselten Daten nur von PGP-Versionen ab 5.0 aufwärts (und OpenPGP-kompatiblen Programmen) wieder entschlüsselt werden. Benutzerinnen von PGP vor Version 5.0 können diese Daten nicht wieder entschlüsseln.

Wenn Ihr Standardschlüssel ein DSS/ElGamal-Schlüssel ist und Sie Daten mit dem RSA-Schlüssel der Empfängerin verschlüsseln, geschieht bei eingeschalteter Option `Always encrypt to default key` genau diese Verschlüsselung mit zwei verschiedenen Schlüsselarten, d. h. eine Empfängerin, die eine ältere Version von PGP als 5.0 benutzt, kann die Daten nicht entschlüsseln.

PGP zeigt Ihnen eine entsprechende Meldung an, wenn dies geschieht (Abb. 21.1). Um sicher zu gehen, daß Sie keine unlesbaren Daten versenden, sollten Sie diese Option im allgemeinen abgeschaltet lassen oder als Default-Schlüssel einen RSA-Schlüssel setzen. Wenn Ihr Standardschlüssel ein RSA-Schlüssel ist oder wenn Sie ganz sicher sind, keine Daten mit Benutzern von PGP-Versionen vor 5.0 auszutauschen, können Sie diese Option aktivieren.

Abbildung 21.1: Warnung: Eventuell gibt es beim Entschlüsseln Probleme

Anwendungsbeispiel:
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 die Absenderin sehr viele E-Mails versendet, kann sie sich nicht mehr erinnern, was sie dem Empfänger geschrieben hat. In diesem Fall kann die Absenderin die Daten nach Eingabe ihres Mantras wieder entschlüsseln, so daß der originale Text wieder vorliegt.

21.1.1.2. Entschlüsselungs-Mantra bereithalten für . . . (Cache decryption passphrases for . . .)

Standardmäßig ist diese Option aktiviert; die voreingestellte Haltezeit beträgt 2 Minuten.

Diese Option bewirkt, daß das Mantra nach der Eingabe für die angegebene Zeit im Arbeitsspeicher gehalten wird. Wenn innerhalb dieser vorgegebenen Zeit eine weitere Entschlüsselung vorgenommen wird, muß es nicht erneut eingegeben werden.

Die Bequemlichkeit wird hier durch ein Sicherheitsrisiko erkauft. Wenn das Mantra bekannt ist, kann jemand, der Zugang zu Ihrem privaten Schlüssel hat, alle Daten mitlesen, die an Sie verschlüsselt wurden – und genau das möchten Sie mit der Verschlüsselung ja vermeiden. Bei Betriebssystemen wie Microsoft Windows werden häufig Daten aus dem Arbeitsspeicher auf die Festplatte ausgelagert, wenn andere Programme Arbeitsspeicher benötigen. Es ist daher nicht auszuschließen, daß Ihr Mantra auf der Festplatte abgespeichert wird.∞ Daher sollten Sie diese Option entweder abschalten oder aber zumindest die Zeit, die das Mantra im Speicher gehalten wird, kurz halten. Wie kurz, hängt von Ihren persönlichen Umständen ab. Wenn Sie Ihren Rechner alleine benutzen und sowieso immer herunterfahren und ausschalten, wenn Sie Ihren Arbeitsplatz verlassen, kann die Zeitspanne natürlich größer sein als wenn Sie einen Rechner mit anderen Menschen teilen und am Ende noch wie an einem Kassensystem immer nur kurz etwas tippen und dann jemand anderen wieder an das Gerät lassen.

21.1.1.3. Signatur-Mantra bereithalten für . . . (Cache signing passphrases for . . .)

Standardmäßig ist diese Option ausgeschaltet.

Diese Option tut im wesentlichen dasselbe wie die eben besprochene; wenn Sie verschiedene Mantras für verschiedene Schlüssel zum Entschlüsseln und Signieren verwenden, bezieht diese Option sich aber nur auf das zum Signieren nötige Mantra. Die Überlegungen aus dem vorangegangenen Kapitel gelten auch hier.

∞ Windows NT bietet einen Systemaufruf, der das verhindern soll – leider scheint dieser Aufruf aber ohne Funktion zu bleiben.

21.1.1.4. Anzeige der Empfänger bei Verschlüsselung an unzureichend beglaubigte Schlüssel (Show recipients when encrypting to marginally valid Keys)

Standardmäßig ist diese Option ausgeschaltet.

Diese Option ist nur bei PGP 5.0i verfügbar. Wenn angewählt, soll Sie bewirken, daß PGP jedesmal eine Warnmeldung ausgibt, wenn an eine Empfängerin verschlüsselt wird, deren Schlüssel nicht ausreichend beglaubigt sind. Die Option ist leider funktionslos, das Verhalten des Programms unterscheidet sich nicht, wenn die Option an- oder abgeschaltet wird – offensichtlich ein Programmierfehler.

21.1.1.5. Kommentar-Block (Comment block) (optional)

Standardmäßig ist kein Kommentar eingegeben.

Diese Option ist nur bei PGP ab Version 5.5 verfügbar, nicht in Version 5.0. Wenn hier ein Text eingegeben wird, wird der eingegebene Text als Kommentar nach der Kopfzeile in die verschlüsselte Daten bzw. in den von PGP erzeugten Signaturblock eingefügt. Dies erfolgt bei allen verschlüsselten und signierten Daten, die im Textformat gespeichert werden.

21.1.2. Schlüsselerzeugungs-Voreinstellungen (Key Generation Preferences)

21.1.2.1. Schnellere Schlüsselerzeugung (Faster Key Generation)

Standardmäßig ist diese Option aktiviert.

Diese Option wurde in Abschnitt 17.1 auf Seite 146 bereits ausführlich behandelt. Sie sollten diese Option abschalten, bevor Sie PGP einen neuen DSS/ElGamal-Schlüssel erzeugen lassen. Für RSA-Schlüssel, die Sie aber nur mit PGP 5.5.3i und 6.5i erzeugen können, ist die Einstellung wirkungslos.

21.1.3. Grundeinstellungen Dateien löschen (File Wiping Preferences)

Diese Funktionen stehen unter PGP 5.0i nicht zur Verfügung, da die Version 5.0i nicht über ein Hilfsprogramm zum Löschen von Dateien verfügt. Nähere Erläuterungen finden Sie in Abschnitt 5.4 auf Seite 37.

21.1.3.1. Vor Überschreiben von Dateien Warnung anzeigen (Display wipe confirmation dialog (PGP 5.5.3i)) (Warn before wiping (PGP 6.0i))

Standardmäßig ist diese Option aktiviert.

PGP besitzt seit Version 2.3a, in den Windows-Versionen seit 5.5.3i eine eingebaute Dateilösch- und -überschreibungsfunktion. Hintergrund ist, daß die Betriebssysteme MS-DOS und die diversen Windows-Varianten (wie die meisten anderen Betriebssysteme auch) eine Datei beim normalen Löschen nicht wirklich von einem Datenträger entfernt, so daß der Inhalt so lange rekonstruierbar bleibt, bis er zufällig durch andere Daten überschrieben wird. Selbst dann könnte der vorherige Inhalt, entsprechende technische Möglichkeiten vorausgesetzt, unter Umständen wieder rekonstruiert werden.

Die Dateilösch- und -überschreibungsfunktion von PGP überschreibt die Daten auf dem Datenträger mit Zufallsdaten, anstatt nur – wie das Betriebssystem – die von den Daten belegten Speicherplätze wieder als verfügbar zu markieren. Dadurch wird eine Rekonstruktion der Daten für normale Anwender unmöglich und selbst für Speziallabors sehr schwierig.

Die Funktion hat allerdings einen echten Schönheitsfehler: Sie läßt den Verzeichniseintrag der gelöschten Datei unberührt. Das bedeutet, daß auf der Festplatte immer noch im Klartext steht, wie die Datei hieß (mit Ausnahme des ersten Buchstabens), wie groß sie war, wann sie das letzte Mal geändert wurde und in welcher Zuordnungseinheit auf dem Datenträger die dazugehörigen Daten beginnen. Zum einen ist das schon mehr Information, als man vielleicht in fremden Händen wissen möchte, zum anderen sind es genau die Informationen, die benötigt werden, um einen Rettungsversuch doch noch relativ aussichtsreich zu machen. Kurz gesagt: Echte Paranoiker und Menschen, die ganz sicher gehen müssen, daß Ihre Daten niemand anderem zugänglich werden, besorgen sich für diesen Zweck bessere Werkzeuge.

Für den normalen Gebrauch, wenn Sie also z. B. vermeiden möchten, daß Ihre Chefin beim nächsten Aufräumen auf der Festplatte zufällig über Ihre Bewerbung für eine Stelle bei der Konkurrenz stolpert, reicht die Funktion allerdings sicherlich aus. Die Dateiüberschreibungs-Funktion sollte mit Vorsicht angewandt werden, versehentlich damit gelöschte Daten sind für normale Anwender (auch mit Edel-Spezial-Programmen) endgültig verloren.

Die Option bewirkt, daß vor dem Überschreiben und Löschen von Dateien mit der PGP-Dateilösch-Funktion ein Dialogfenster angezeigt wird, in dem die betroffenen Dateien aufgelistet werden und der Benutzer die Löschung nochmals bestätigen muß. Es empfiehlt sich, diese Option aktiviert zu lassen.

21.1.3.2. Anzahl der Überschreibungs-Vorgänge (Number of passes)

Standardwert: 8.

Diese Option ist nur bei PGP 6.0i verfügbar. Hier kann angegeben werden, wie oft die Dateilösch- und -überschreibungsfunktion eine Datei überschreibt. Je häufiger dies geschieht, umso länger dauert der Vorgang. Andererseits sinkt die Chance, die Daten aus dem Hintergrundrauschen der Festplatte rekonstruieren zu können, mit jedem zusätzlichen Überschreiben.

Standardmäßig steht die Anzahl der Überschreibungsvorgänge auf 8. Wer mehr benötigt, hat wirklich sensible und für Ausspähung gefährdete Daten und sollte sich wegen des Problems mit den Verzeichniseinträgen, das im vorangegangenen Abschnitt angesprochen wurde, lieber ein anderes Programm für diesen Zweck besorgen.

21.2. Registerkarte Dateien (Key Files (PGP 5.0i)) (Files (PGP 5.5.3i und PGP 6.0i))

In diesem Register werden die Pfade zu Ihren Schlüsselbund-Dateien angegeben. Bitte lesen Sie hierzu auch in die Abschnitte 7.4 auf Seite 65 und 7.2 auf Seite 62.

In allen zwei bzw. drei Eingabefeldern können Sie die Pfade entweder direkt eingeben oder über die Schaltfläche `Browse` auch den Windows-Verzeichniswechsel-Dialog aufrufen und die jeweilige Datei dort anwählen.

21.2.1. Öffentlicher Schlüsselbund (Public Key Ring File)

Hier können Sie den Pfad zu der Datei angeben, die Ihren öffentlichen Schlüsselbund enthält. Diese Schlüssel können Sie frei an die Mensch-

heit verteilen, sie müssen nicht vor Lesen geschützt werden und können daher ruhig auf der Festplatte verbleiben.

21.2.2. Privater Schlüsselbund (Private Key Ring File)

Hier können Sie den Pfad zu der Datei angeben, die Ihren privaten Schlüsselbund enthält. Diese sollte niemand außer Ihnen in die Finger bekommen, daher sollten Sie die Datei `secring.skr` von einem Wechselmedium, beispielsweise einem ZIP-Medium, einer CD oder notfalls einer Diskette benutzen, wenn Dritte Zugriff auf Ihren Rechner haben. Das mag paranoid erscheinen, aber Windows 95/98 bieten keinerlei Schutz vor den Mitbenutzern des Rechners – bitte lesen Sie hierzu auch Kapitel 5.8 auf Seite 43. Beachten Sie aber dabei unbedingt den Hinweis auf die Sicherungskopien in Kapitel 7.4 auf Seite 65!

21.2.3. Zufallszahlengenerator-Startwerte-Datei (Random Seed File)

Diese Einstellungsoption steht in PGP 5.0i nicht zur Verfügung, denn bei PGP 5.0i muß sich die Datei `randseed.bin` immer im Programmverzeichnis von PGP 5.0i befinden.*

Hier können Sie den Pfad zu der Datei angeben, die die Startwerte für den Zufallszahlen-Generator enthält, der die Zufallszahlen für die Sitzungs-Schlüssel erzeugt. Diese Datei kann auch auf der Festplatte verbleiben, da sie erstens verschlüsselt wird und sich zweitens bei jeder Benutzung von PGP mit zufälligen Werten wieder ändert, so daß ein Angriff darauf sehr schwer und nicht sehr erfolgversprechend ist.

* Das ist bei der Kommandozeilenversion ebenfalls der Fall, obwohl dort eine Option existiert, die Datei an anderer Stelle zu suchen – dann wird sie allerdings nur verändert, aber ansonsten nicht benutzt.

21.3. Registerkarte E-Mail

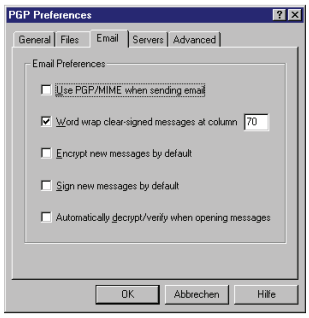


Abbildung 21.2: Einstellungen E-Mail (5.5.3i)

Die hier aufgeführten Optionen betreffen die Erweiterungen von E-Mailprogrammen, sogenannte Plugins, die die Funktionen von PGP für Windows direkt in die E-Mailprogramme integrieren. Für eine Benutzung von PGP ohne diese Plugins über die Windows-Zwischenablage, mit den PGP-Erweiterungen des Windows-Explorers oder über `PGPtools` haben die Einstellungen keine Bedeutung.

Nicht alle Funktionen stehen in allen erhältlichen Plugins zur Verfügung.

21.3.1. PGP/MIME zum Verschicken von E-Mail benutzen (Use PGP/MIME when sending email)

Standardmäßig ist diese Option ausgeschaltet.

Wenn diese Option aktiviert ist, benutzt das jeweilige Plugin für Ihr E-Mailprogramm (so vorhanden) automatisch den Standard PGP/MIME zum Verschicken von E-Mails. Damit werden alle Mails automatisch an die jeweiligen Empfänger verschlüsselt und von Ihnen signiert, ohne daß Sie das jedesmal angeben müssen. Darüber hinaus wird das Datenformat PGP/MIME für den Versand der E-Mail benutzt. Diese Option sollten Sie nicht aktivieren, wenn Sie nicht sicher sind, ob Ihre Kommunikationspartner ausnahmslos E-Mailprogramme benutzen, die dieses Format auch unterstützen. Davon können Sie zumindest derzeit aber nicht ausgehen.

21.3.2. Zeilenumbruch bei signierten Klartext-Nachrichten in Spalte . . . (Word wrap clear-signed messages at column . . .)

Standardmäßig ist diese Option aktiviert, die Zeilenbreite steht auf 70 Zeichen.

Mit dieser Option wird eingestellt, ab welcher Spalte (Anzahl Zeichen) der Zeilenumbruch bei E-Mails erfolgen soll, die unverschlüsselt verschickt, aber mit einer Signatur versehen werden.

Bedingt durch die Technik der Signatur darf an einem Text nach der Signierung nichts mehr geändert werden. Manche E-Mail-Programme brechen eingegebenen Text aber nicht selbständig nach den üblichen 70-72 Zeichen um. Es ist generell sehr zu empfehlen, diesen Umbruch einzuschalten (so denn möglich). Diese Einstellung veranlaßt PGP, die nötigen Zeilenumbrüche notfalls vor einer Klartext-Unterschrift einzufügen. Mit veränderlichen Zeilenumbrüchen oder Zeilenumbrüchen, die das E-Mail-Programm erst nach dem PGP-Aufruf einfügt, würde die Signatur nicht mehr zum Text passen, wäre also ungültig.

Der Zeilenumbruch sollte so gewählt werden, daß er auch auf E-Mail- und News-Programmen, die selbst kein „word wrap“ ausführen, auch nach mehrmaligem Zitieren noch lesbar ist. Allgemein üblich ist (und erwartet wird) ein Zeilenumbruch nach 70-72 Zeichen, also sollten Sie die Voreinstellung von 70 Zeichen ruhig stehenlassen. Von einem Anwender wurde berichtet, daß sein Outlook98 nicht mit PGP zusammenarbeitete, wenn er den Zeilenumbruch nicht oder auf zu große Werte einstellte.

21.3.3. Neue Nachrichten standardmäßig verschlüsseln (Encrypt new messages by default)

Standardmäßig ist die Option ausgeschaltet.

Wenn Sie ein E-Mailprogramm benutzen, für das ein PGP-Plugin installiert ist und Sie diese Option aktiviert haben, wird von PGP versucht, alle ausgehenden Mails automatisch an den Empfänger zu verschlüsseln.

Wenn Sie bei einer einzelnen E-Mail nicht möchten, daß sie verschlüsselt wird, müssen Sie das Verschlüsseln für diese Nachricht vor dem Versenden in Ihrem E-Mailprogramm abschalten.

Wenn Sie diese Option aktiviert haben und eine Nachricht an eine Empfängerin schicken, von der Sie keinen Schlüssel in Ihrem Schlüsselbund haben, zeigt Ihnen PGP das Empfänger-Auswahl-Fenster an (siehe auch im Kapitel 20.1 auf Seite 231), wo Sie von Hand einen Empfänger auswählen können, an den Sie verschlüsseln möchten.⊘ Wenn Sie die E-Mail aber gar nicht verschlüsseln können, weil die Empfängerin PGP nicht benutzt oder Sie die Schlüssel der Empfänger nicht haben, so

⊘ Bei PGP 5.0i versucht das Programm unter Umständen zuerst, abhängig von Ihren Einstellungen auf der Registerkarte Key Server, eine Verbindung zum voreingestellten Standard-Keyserver im Internet aufzubauen, um sich den fehlenden Schlüssel dort zu besorgen. Bitte lesen Sie hierzu den Abschnitt 21.4 auf Seite 251.

müssen Sie diesen Dialog beenden, dann die Verschlüsselung in Ihrem E-Mailprogramm abschalten und die Nachricht nochmals verschicken.

Da dies ein wenig lästig werden kann, wenn der größte Teil Ihrer Nachrichten nicht verschlüsselt werden soll, ist das Einschalten der standardmäßigen Verschlüsselung nur sinnvoll, wenn die meisten Ihrer Nachrichten verschlüsselt werden. Ansonsten tun Sie sich leichter, die Verschlüsselung jeweils für diejenigen Nachrichten extra anzuschalten, für die sie verwendet werden soll.

Andererseits ist diese Option ein gutes Mittel dagegen, das Verschlüsseln zu vergessen. Letztlich ist es also Geschmackssache, ob Sie die Option ein- oder ausschalten. Beachten Sie aber bitte unbedingt, daß es für manche Empfänger (beispielsweise in Ländern mit Kriegsrecht) lebensbedrohlich sein kann, verschlüsselte Nachrichten zu empfangen! Wenn Sie also jemand bittet, unverschlüsselt zu schreiben, sollten Sie das auf jeden Fall tun.

21.3.4. Neue Nachrichten standardmäßig signieren (Sign new mesages by default)

Standardmäßig ist diese Option ausgeschaltet.

Wenn Sie ein E-Mailprogramm benutzen, für das ein PGP-Plugin installiert ist und wenn Sie diese Option aktiviert haben, werden von PGP alle ausgehenden Mails automatisch signiert. Für das Signieren benötigen Sie nur Ihren eigenen Schlüssel und die Nachricht kann auch von Empfängerinnen gelesen werden, die PGP nicht benutzen.

Die Kryptographie lebt nicht zuletzt davon, daß möglichst viele Leute sie benutzen, um von dem falschen Image wegzukommen, etwas „zu verbergen zu haben“. Es kann also nicht schaden, wenn möglichst viele Menschen dokumentieren, daß sie Kryptographie einsetzen, um Ihre Privatsphäre zu schützen. Ein gutes Mittel, dafür ein wenig Werbung zu machen, ist alle Nachrichten zu signieren. Sie sollten diese Option daher ruhig aktivieren.

21.3.5. Automatisch entschlüsseln/Signatur prüfen beim Öffnen von Nachrichten (Automatically decrypt/verify when opening messages)

Diese Option steht in PGP 5.0i nicht zur Verfügung. Standardmäßig ist die Option ausgeschaltet.

Wenn Sie in Ihrem E-Mailprogramm eine Nachricht öffnen, die mit PGP verschlüsselt oder mit PGP signiert wurde, so bekommen Sie normalerweise den verschlüsselten Text bzw. den Text mit dem Signaturblock angezeigt und müssen die Prüfung der Signatur bzw. das Entschlüsseln von Hand einleiten, indem Sie die entsprechende Schaltfläche in Ihrem E-Mailprogramm anklicken.

Wenn Sie ein E-Mailprogramm benutzen, für das ein PGP-Plugin installiert ist und diese Option aktiviert haben, so leitet PGP die Entschlüsselung bzw. die Prüfung der Signatur beim Öffnen der Nachricht automatisch ein, Sie müssen sie also nicht jedes Mal von Hand anstoßen. Sie müssen dann nur noch Ihr Mantra eingeben; sollte das Mantra noch im Speicher sein (s. o. unter Kapitel 21.1.1.2 auf Seite 243), dann werden Ihre Nachrichten sofort im Klartext angezeigt. Sie sollten den Rechner also nicht unbeaufsichtigt lassen, während das Mantra noch im Speicher ist!

Da sich PGP nicht daran stört, wenn eine geöffnete Nachricht gar nicht verschlüsselt oder signiert ist, sollten Sie diese Option ruhig aktivieren, sie steigert den Bedienungskomfort Ihres E-Mailprogramms mit PGP deutlich.

21.4. Registerkarte Keyserverprotect (Key Server (PGP 5.0i)) (Servers (PGP 5.5.3i und PGP 6.0i))

In diesem Register können Sie die Liste der über HTTP+ oder LDAP+ anzusprechenden Keyserver im Internet verwalten, die Sie für PGP benutzen möchten. Über diese Server können Sie Ihre öffentlichen Schlüssel verbreiten, Sie können öffentliche Schlüssel anderer Leute bekommen, Sie können kundtun, daß Sie Ihren Schlüssel zurückziehen möchten, daß er also nicht mehr verwendet werden darf, und so weiter.

Nähere Informationen und Zugriffsmöglichkeiten erhalten Sie über PGP.NET oder über PGPI.COM.

Neben den HTTP- und LDAP-Keyservern gibt es auch „klassische“ Keyserver wie z. B. pgp-Public-Keys@informatik.uni-hamburg.de, die per

+ HTTP, LDAP und LDAPS sind verschiedene Protokolle, mit denen Keyserver über Internet bedient werden können. Für uns ist hier nur wichtig, daß LDAPS eine verschlüsselte, authentisierte Kommunikation bietet, so daß es grundsätzlich sogar möglich ist, damit Schlüssel und Schlüsselteile wieder vom Server zu löschen – das funktioniert aber nur bei einzelnen Servern und wird nicht an andere weitergegeben, was den Nutzen der Möglichkeit sehr stark einschränkt.

E-Mail bedient werden. Diese Keyserver können Sie daher auch ohne aktivierte Internet-Anbindung abfragen; die meisten von ihnen arbeiten aber nur mit RSA-Schlüsseln.

Eine E-Mail an diese Adresse mit dem Betreff `HELP` ohne Nachrichten-Text bringt diesen Keyserver z. B. dazu, Ihnen eine E-Mail mit einem Hilfetext und einer Bedienungsanleitung des Keyservers zurückzusenden. Diese Keyserver sind jedoch meist auf RSA-Schlüssel spezialisiert und nehmen keine DSS/ElGamal-Schlüssel an.

21.4.1. PGP 5.0i

In den Grundeinstellungen von PGP 5.0i können Sie nur einen einzigen Internet-Keyserver angeben. Grundeinstellung ist der Server `horowitz.surfnet.nl`, Port 11371. Die Porteinstellung sollten Sie unverändert lassen, alle `HTTP`-basierten PGP-Keyserver „lauschen“ auf Port 11371, lediglich einige `LDAP`-basierte Keyserver arbeiten auf Port 389. Auch den voreingestellten Server `horowitz.surfnet.nl` können Sie als Voreinstellung belassen, er ist noch in Betrieb.

Wenn die Option `Automatically retrieve unknown keys` (unbekannte Schlüssel automatisch besorgen) aktiviert ist, versucht PGP 5.0i jedesmal eine Verbindung zum voreingestellten Keyserver aufzubauen, wenn es einen Schlüssel benötigt, der nicht im öffentlichen Schlüsselbund ist (z. B. beim Senden an eine E-Mail-Adresse, zu der Sie keinen passenden Schlüssel in Ihrem Schlüsselbund haben). PGP versucht dann, den fehlenden Schlüssel vom Keyserver zu bekommen. Dies funktioniert natürlich nur dann, wenn Ihre Internet-Verbindung gerade aktiv ist.

21.4.2. PGP 5.5.3i

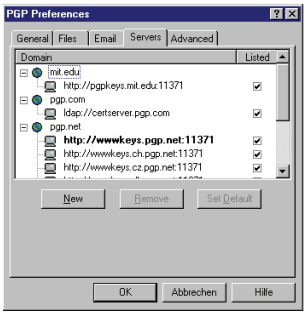


Abbildung 21.3: Liste der Keyserver (5.5.3i)

Bei PGP 5.5.3i können Sie nahezu beliebig viele Keyserver in Ihrer Suchliste halten (Abb. 21.3). Schon von Haus aus bringt PGP 5.5.3i eine lange Liste an eingetragenen Keyservern mit. Den Keyservern wird ein bestimmter Adressbereich („Domain“) zugewiesen, für den sie zuständig sind. Das heißt, wenn Sie nach einer E-Mail-Adresse in der Domain `.de` suchen, wird zuerst der Keyserver angesprochen, den Sie dafür angegeben haben. Ist für die gesuchte Adresse kein zuständiger Keyserver angegeben, wird der Standard-Keyserver benutzt. Im Allgemeinen halten alle Keyserver denselben Bestand an Schlüsseln vor, aber es kann z. B. vorkommen, daß Ihre Firma einen eigenen Server für die Schlüssel der Mitarbeiter betreibt.

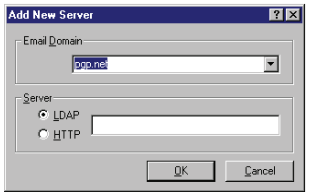


Abbildung 21.4: Neuen Keyserver einrichten (5.5.3i)

Mit der Schaltfläche `New` (Neu) können Sie neue Keyserver zu Ihrer Liste hinzufügen. In dem dann erscheinenden Fenster (Abb. 21.4) geben Sie im oberen Feld unter `Email Domain` den Adressbereich ein, für den der Server zuständig sein soll. Wenn Sie einen Server für eine Domain eingeben wollen, für die Sie bereits einen Server angegeben hatten, können Sie die Domain auch durch Klicken auf das Listenfeld neben dem Eingabebereich aus den vorhandenen Domains auswählen. Im unteren Bereich geben Sie über das Auswahlfeld an, ob der Server über das `LDAP`◁- oder das `HTTP`◁-Protokoll angesprochen wird (das sollten Sie vorher in Erfahrung bringen). Daneben geben Sie die Internet-Adresse des Servers in das freie Feld ein, also einen Rechnernamen (wie z. B. `horowitz.surfnet.nl`) oder IP-Adresse gefolgt von einem Doppelpunkt und der Port-Adresse (bei `HTTP`-basierten PGP-Keyservern im allgemeinen der Port 11371, bei einigen `LDAP`-basierten PGP-Keyservern auch Port 389). Mit `OK` fügen Sie den neuen Server der Liste hinzu und schließen das Fenster.

Wenn Sie einen Keyserver endgültig aus der Liste löschen möchten,

◁ Siehe Fußnote+ auf Seite 251.

z. B. weil er nicht mehr ansprechbar ist, dann müssen Sie den entsprechenden Server in der Liste mit einem Mausklick markieren und danach auf die Schaltfläche `Remove` (Entfernen) klicken. PGP fragt sicherheitshalber noch einmal nach, ob Sie den Server auch wirklich aus der Liste löschen möchten, hier müssen Sie ggf. mit `Ja` bestätigen. Wenn Sie den letzten Keyserver für eine Domain entfernt haben, wird die Domain automatisch aus der Liste der Keyserver gelöscht. Wenn danach ein Schlüssel für eine E-Mail-Adresse aus dieser Domain gesucht wird, ist wieder der Standard-Keyserver zuständig.

Wenn Sie einen anderen als den derzeitigen Standard-Keyserver (Erkennbar am fettgedruckten Namen) zum Standard-Keyserver machen möchten, so markieren Sie den Neuen durch einem Mausklick und Klicken danach auf die Schaltfläche `Set Default`.

Wenn der Punkt `Listed` (Angezeigt) hinter einem Keyserver aktiviert ist, dann wird der betreffende Keyserver in den Auswahlfeldern der Funktionen von `PGPkeys` zum Senden an und zum Durchsuchen von Keyservern angezeigt, sonst fehlt der betreffende Keyserver in diesen Anzeigefeldern.

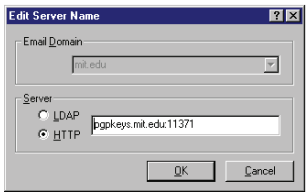


Abbildung 21.5: Servereinstellungen ändern (5.5.3i)

Wenn Sie die Einstellungen eines bestehenden Keyservers verändern möchten, dann klicken Sie einmal mit der rechten Maustaste darauf und wählen den (einzigen) Punkt `Edit` aus dem erscheinenden Menü. Das darauf erscheinende Fenster (Abb. 21.5) enthält die selben Felder wie das Fenster zum Anlegen eines neuen Keyservers in der Liste (s. o.), allerdings können Sie nur noch das verwendete Protokoll und den Namen bzw. die IP-Adresse ändern, die Angabe für die Domain-Zuständigkeit kann hier nicht mehr geändert werden. Wenn Sie die Zuständigkeit eines Keyservers ändern möchten, müssen Sie ihn unter der neuen Domain neu anlegen und ggf. danach unter der alten löschen.

21.4.3. PGP 6.0i

Im großen und ganzen wird die Keyserverliste wie bei PGP 5.5.3i bedient, die grundlegenden Funktionen zum Hinzufügen, Löschen usw. sind dieselben. Allerdings sind einige Funktionen hinzugekommen (Abb. 21.6) und die graphische Gestaltung der Eingabemasken ist anders.

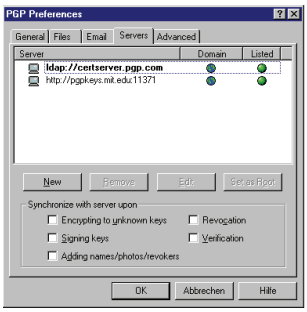


Abbildung 21.6: Server-Einstellungen (6.0i)

PGP 6.0i bringt nur zwei Keyserver als Grundausstattung in der Liste mit.

Mit einem Klicken auf die Schaltfläche `New` kommen Sie zum Fenster für die Eingabe neuer Keyserver (Abb. 21.7). Hier geben Sie zuerst über das Listenfeld ganz oben das Protokoll ein, mit dem der Server arbeitet. Zur Auswahl stehen `HTTP`,◁ `LDAP`◁ und `LDAPS`,◁ eine `LDAP`-Implementation mit SSL, also mit verschlüsselter und authentifizierter Übertragung.

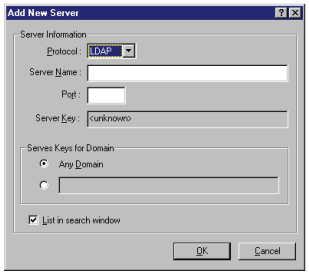


Abbildung 21.7: Einrichten eines neuen Keyservers (6.0i)

Über `LDAPS` können Sie z. B. nach Authentifizierung über den eigenen privaten Schlüssel, eigene Schlüssel selbst auf dem Keyserver deaktivieren oder löschen. Für die öffentlichen Keyserver ist das insofern von beschränktem Nutzen, als diese Einstellungen nicht an andere Keyserver weitergegeben werden (können). Für einen firmeneigenen Keyserver ist es aber eine lohnende Neuerung.

`ldaps://certserver.pgp.com` ist z. B. ein mit PGP 6.0i kompatibler Keyserver, der diese Operationen zuläßt.

Im Feld `Server name` geben Sie wie gehabt den Namen oder die IP-Adresse Ihres gewünschten Keyservers ein. Im Feld `Port` geben Sie den Port ein, an dem der Server „lauscht“. Dies ist für PGP-Keyserver, die mit dem `HTTP`-Protokoll arbeiten, im allgemeinen Port 11371, für `LDAP` und `LDAPS`-Keyserver häufig auch der Port 389.

Im Feld `Server Key` können Sie nichts eingeben. Hier soll der Schlüssel eines `LDAPS`-Keyservers stehen, der zur Authentifizierung einer Verbindung benutzt wird.

◁ Siehe Fußnote+ auf Seite 251.

Im Feld `Serves keys for Domain` können Sie angeben, für welchen Adressbereich (für welche Domain) der Server benutzt werden soll. Zur Auswahl steht entweder der Listenpunkt `Any Domain`, wenn der Keyserver für E-Mail-Adressen aus allen Domains zuständig sein soll, oder Sie können den zweiten Punkt aktivieren und eine bestimmte Domain im Textfeld angeben.

Wenn der Punkt `List in search window` aktiviert ist, dann wird der betreffende Keyserver in den Auswahlfeldern der Funktionen von `PGPkeys` zum Senden an und zum Durchsuchen von Keyservern angezeigt, sonst fehlt der Server in diesen Anzeigen.

Durch Klicken auf die Schaltfläche `Remove` entfernen Sie einen Keyserver aus der Liste; die Schaltfläche können Sie nur auswählen, wenn Sie vorher einen Keyserver aus der Liste durch einen Mausklick ausgewählt haben. Vor dem Löschen kommt noch eine Sicherheitsabfrage, die Sie zum Löschen mit `Ja` bestätigen müssen.

Durch Klicken auf die Schaltfläche `Edit` können Sie die Einstellungen eines vorhandenen Keyservers ändern; die Schaltfläche können Sie nur auswählen, wenn Sie vorher aus der Liste einen Keyserver durch einen Mausklick ausgewählt haben. Das Fenster ist inhaltlich das selbe wie bei der Eingabe eines neuen Servers (s. o.). Im Gegensatz zu PGP 5.5.3i können Sie hier auch nachträglich den Adressbereich ändern, für die der Server zuständig sein soll.

Den Standard-Keyserver können Sie ändern, indem Sie einen anderen als den derzeitigen Standard-Keyserver in der Liste durch Mausklick markieren und dann die Schaltfläche `Set as Root` anklicken. Wenn der derzeitige Standard-Keyserver oder kein Keyserver markiert ist, ist diese Schaltfläche nicht wählbar.

21.4.3.1. Schlüssel mit Keyserver synchronisieren bei . . . (Synchronize with server upon)

Im unteren Teil der Keyserver-Grundeinstellungen können Sie angeben, bei welcher Gelegenheit PGP automatisch versuchen soll, die Schlüsseldaten mit den angegebenen Keyservern abzustimmen, also sie von dort anzufordern und ggf. nach der Bearbeitung die geänderten Daten wieder dorthin zurückzuschicken.

Alle diese Einstellungen funktionieren nur bei aktivierter Internet-Verbindung.

Verschlüsselung an unbekannten Schlüssel (Encrypting to unknown keys)

Wenn eine Nachricht an Empfänger verschlüsselt werden soll, von denen kein Schlüssel im öffentlichen Schlüsselbund gefunden werden kann, dann wird der entsprechende Schlüssel vom Keyserver angefordert, wenn diese Option aktiviert ist. So soll erreicht werden, daß auch an Kommunikationspartnerinnen ohne großen Aufwand verschlüsselte Daten geschickt werden können, deren Schlüssel Sie noch nicht haben.

Diese Option entspricht der Einstellung `Automatically retrieve unknown keys` auf der Key Server-Registerkarte von PGP 5.0i.

Signieren eines Schlüssels (Signing keys)

Wenn ein Schlüssel signiert werden soll, wird zuerst vom Keyserver die aktuellste Version des zu signierenden Schlüssels angefordert, wenn diese Option aktiviert ist. Nach erfolgter Signatur wird der Schlüssel mit Ihrer neu hinzugefügten Unterschrift wieder an den Keyserver zurückgeschickt.

Damit soll sichergestellt werden, daß der zu signierende Schlüssel z. B. nicht in der Zwischenzeit zurückgezogen wurde und daß die Signaturen, die das Vertrauensnetzwerk bilden sollen, sich möglichst weit verbreiten (zum Vertrauensnetz lesen Sie bitte näheres in den Abschnitten 7.1.2 auf Seite 59 und 7.3 auf Seite 63).

Hinzufügen von Namen, Photos und Rückrufern (Adding names/photos/revokers)

Wenn einem Schlüssel ein zusätzlicher Benutzername, ein Photo oder ein Rückrufer hinzugefügt werden soll, wird zuerst vom Keyserver die aktuellste Version des zu ergänzenden Schlüssels angefordert, wenn diese Option aktiviert ist. Nach erfolgter Änderung wird der Schlüssel mit den neu hinzugefügten Komponenten wieder an den Keyserver zurückgeschickt. Der Grund ist derselbe wie im vorangehenden Punkt; den Schlüssel vom Keyserver zu holen, ist allerdings nur beschränkt sinnvoll – es handelt sich schließlich um Ihren eigenen Schlüssel.

Zurückrufen (Revocation)

Wenn ein Schlüssel zurückgerufen werden soll, wird zuerst vom Keyserver die aktuellste Version des zurückzurufenden Schlüssels angefordert, wenn diese Option aktiviert ist. Nach erfolgtem Rückruf wird der nun ungültige Schlüssel mit Ihrem neu hinzugefügten Rückruf wieder an den Keyserver zurückgeschickt.

Hiermit soll verhindert werden, daß Rückrufe für Schlüssel abgesetzt werden, die bereits zurückgerufen worden sind. Die Option kann vermutlich bedenkenlos abgeschaltet werden.

Prüfung einer Signatur (Verification)

Wenn eine Signatur von Absendern geprüft werden soll, von denen kein Schlüssel im öffentlichen Schlüsselbund gefunden werden kann, dann wird der entsprechende Schlüssel vom Keyserver angefordert, wenn diese Option aktiviert ist. So soll erreicht werden, daß Sie ohne großen Aufwand Signaturen von Leuten prüfen können, über deren Schlüssel Sie bis dahin noch nicht verfügt haben.

21.5. Registerkarte Fortgeschrittene Einstellungen (Advanced)

Diese Registerkarte (Abb. 21.8) und die auf ihr enthaltenen Funktionen stehen unter PGP 5.0i nicht zur Verfügung.

21.5.1. Verschlüsselung (Encryption)

21.5.1.1. Erlaubte Algorithmen (Enabled/Allowed Algorithms)

Standardeinstellung: Alle unterstützen Algorithmen.

Unter dem Punkt `Enabled algorithms` (PGP 5.5.3i) bzw. `Allowed algorithms` (PGP 6.0i) können Sie auswählen, welche Verschlüsselungsalgorithmen PGP verwenden darf: CAST, TripleDES und IDEA können jeweils einzeln aktiviert und abgeschaltet werden. Es sollten alle drei aktiviert bleiben (das ist die Standardeinstellung). Die Auswahl ist dafür gedacht, daß ein bestimmter Algorithmus stillgelegt werden kann, falls er sich in Zukunft als unsicher herausstellen sollte. Für den kommer-

ziellen Einsatz sollte aus lizenzrechtlichen Gründen IDEA abgeschaltet werden, aber die hier besprochenen Programmversionen dürfen ohnehin nicht gewerblich eingesetzt werden.

21.5.1.2. Bevorzugter Algorithmus (Preferred algorithm)

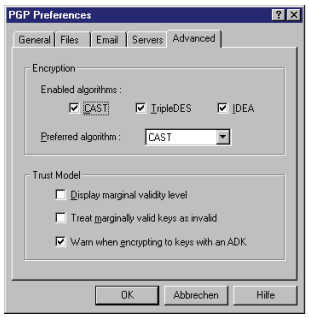


Abbildung 21.8: Fortgeschrittene Einstellungen (5.5.3i)

Welcher Algorithmus letztendlich benutzt wird, legt man über das Listenfeld `Preferred algorithm` fest. CAST ist die Standardeinstellung für DSS/ElGamal-Schlüssel. Wenn Sie TripleDES oder IDEA verwenden möchten, müssen Sie diese Einstellung vor der Schlüsselpaar-Erzeugung ändern (das ist normalerweise nicht notwendig). RSA-Schlüssel arbeiten immer mit dem IDEA-Algorithmus, unabhängig von der Einstellung in dieser Registerkarte.

21.5.2. Vertrauenseinstellungen (Trust Model)

21.5.2.1. Anzeige des Gültigkeits-Niveaus (Display Marginal Validity Level)

Wenn diese Option aktiviert ist, dann wird die Gültigkeit von Schlüsseln im Fenster von `PGPkeys` in drei Stufen angezeigt:

Ungültig Der Schlüssel wurde entweder von niemandem unterschrieben, oder ist nur mit Unterschriften von Schlüsseln versehen, die ebenfalls nicht gültig sind oder deren Inhaber Sie als nicht vertrauenswürdig eingestuft haben.

Halb gültig Der Schlüssel wurde von einem gültigen Schlüssel unterschrieben, dessen Inhaberin Sie als bedingt vertrauenswürdig eingestuft haben.

Gültig Der Schlüssel wurde entweder von Ihnen selbst unterschrieben, oder er ist mit einem gültigen Schlüssel unterschrieben, dessen Inhaberin Sie als vertrauenswürdig eingestuft haben, oder er ist von

mindestens zwei gültigen Schlüsseln unterschrieben, deren Inhaber Sie als bedingt vertrauenswürdig eingestuft haben.

Wenn diese Option nicht eingeschaltet ist, dann gibt es keine Abstufung in der Gültigkeit; ein Schlüssel ist entweder gültig oder nicht. Näheres zu den Gültigkeitsregeln finden Sie in Abschnitt 7.3 auf Seite 63.

21.5.2.2. Nicht vollständig gültige Schlüssel als nicht vertrauenswürdig behandeln (Treat Marginally Valid Keys as Untrusted)

Die Option ist funktionslos, das Verhalten des Programms unterscheidet sich nicht, wenn die Option an- oder abgeschaltet wird. Alle nicht vollständig gültigen Schlüssel als nicht vertrauenswürdig zu behandeln, ist natürlich die Standardvorgehensweise.

21.5.2.3. Bei Verschlüsselung an Schlüssel mit ADK Warnung ausgeben (Warn when encrypting to keys with an ADK)

Wenn diese Option aktiviert ist, wird der ADK-Schlüssel (Additional decryption key – Schlüssel eines Dritten, an den zwangsweise mitverschlüsselt wird) zusätzlich im Schlüsselauswahlfenster angezeigt, er ist mit einem kleinen, gelben Schloßsymbol versehen. Diese Option sollte *unbedingt* eingeschaltet sein.

21.5.2.4. Export-Format (Export Format)

Diese Auswahl-Optionen stehen nur in PGP 6.0i zur Verfügung.

Compatible Beim Export eines Schlüssels wird ein Datenformat verwendet, das frühere Versionen von PGP verwenden können.

Diese Option sollte aktiviert sein, es sei denn, bei der Empfängerin der Schlüsseldaten handelt es sich sicher ebenfalls um eine Benutzerin von PGP 6.0i oder neuer oder GnuPG.

Complete Beim Export eines Schlüssels wird das in Version 6.0 eingeführte Dateiformat verwendet, das unter anderem ein Bild (Photo-ID) im Schlüssel enthalten kann. Außerdem sind sogenannte Rückrufer vorgesehen. Rückrufer sind Dritte, die vom Schlüsselinhaber

berechtigt werden, seinen Schlüssel für ihn zurückzuziehen (z. B. bei Verlust der Schlüsseldaten durch Festplattendefekt kombiniert mit fehlender Sicherungskopie sinnvoll...). Das Format ist nicht kompatibel zu vorherigen PGP-Versionen und sollte daher außerhalb geschlossener Benutzergruppen (noch) nicht benutzt werden.

Teil IV.

Anhang

A. Die vielen PGP-Versionen

Was soll eigentlich das ganze Chaos mit den Versionen 2.3a, 2.6, 2.6MIT, 2.6.1, 2.6ui, 2.6.3i, 2.7, 5.0, 5.0i, 5.5, 6.0, GnuPG usw.?

A.1. Die Versionen 2.x

Bis zur Version 2.3 (2.3a ist ein kleiner Bugfix, d. h. es ist ein Programmierfehler entfernt worden, der Klartext-Unterschriften betraf) war PGP „Guerilla-Freeware“, die außerhalb der USA unter PHILIP ZIMMERMANNs Federführung entstand. Die Verwendung von PGP innerhalb der USA stellte eine Verletzung von Patentrechten dar, da eine eigenständige Implementation des in den USA patentierten Algorithmus RSA verwendet wurde.

Das nächste PGP, das entwickelt wurde (2.4), umging diese Probleme dadurch, daß es sich um eine kommerzielle Version handelte, die von ViaCrypt, Phoenix, Arizona vertrieben wurde. Mit dem Kauf dieser Version erhielt man das Recht, PGP kommerziell einzusetzen, die nötigen Abgaben an PKP und Ascom Tech waren im Preis eingeschlossen.

Um eine in den USA legale Version von PGP zu erstellen, schloß PHILIP ZIMMERMANN ein Abkommen mit PKP und dem MIT, ein PGP zu entwickeln, das die frei verwendbaren RSAREF-Routinen benutzt. Teil dieser Abmachung war eine Änderung der Lizenzbestimmungen PGPs. PGP wird seit der Version 2.5 (die erste MIT-Version) nicht mehr unter den Bestimmungen der General Public License der Free Software Foundation (Stichwort GNU-Projekt) vertrieben, sondern diese Bestimmungen haben Einschränkungen erfahren, die sich zum Teil in den RSAREF-Bedingungen begründen, zum Teil in der Abmachung zwischen PHILIP ZIMMERMANN und PKP.

Kurze Zeit nach der Freigabe der Version 2.5 entstand die Version 2.6. Diese Version ist – wiederum aufgrund eines Abkommens mit o. g. Institutionen – so konstruiert, daß sie seit dem 1. September 1994 Nachrichten, Schlüssel und Unterschriften erzeugt, die mit früheren Ver-

sionen nicht gelesen werden können. Das dient dazu, die Verwendung der (in den USA immer noch gegen Patenrechte verstoßenden) Version 2.3(a) einzudämmen. Um internationale PGP-Kommunikation zu ermöglichen, ohne US-Exportverbote zu verletzen, haben einige Leute sehr bald die nötigen Änderungen am Quelltext veröffentlicht. Diese hat PETER SIMONS in seine Amiga-Version 2.3a.2 eingebaut, wobei die Version 2.3a.3 entstand (die aktuellen Versionen von PGP lassen sich direkt für den Amiga kompilieren) und ein Engländer mit dem Pseudonym MATHEW hat sie in eine MS-DOS-Version 2.3a eingebaut und das Produkt 2.6ui genannt, wobei „ui“ für „unofficial international“ steht. Unofficial deshalb, weil die Version von keinem Programmierer des PGP-Teams abgesegnet ist, international deshalb, weil die Version außerhalb der USA entstanden ist und ohne Probleme in allen Ländern, die Verschlüsselung gestatten, verwendet werden kann – außer in den USA, wo sie gegen Patentrechte verstößt.

Um ihren kommerziellen Kunden eine Kommunikation mit Anwendern der Version 2.6 zu ermöglichen, brachte ViaCrypt die Version 2.7 auf den Markt.

Die letzte offizielle Version der Reihe 2.6.x ist die 2.6.2, ein Bugfix zur 2.6 vom MIT. Diese Version sollte nur innerhalb der USA verwendet werden, da sie RSAREF-Code verwendet, dessen Einsatz außerhalb der USA nicht durch die Lizenzbestimmungen gedeckt ist. Es gibt aber eine Version 2.6.2i (deren Verwendung wir empfehlen), die die Routinen verwendet, die PHIL ZIMMERMANN für PGP 2.0 geschrieben hatte. Weiterhin hat STÅLE SCHUMACHER aus der 2.6.2i die Version 2.6.3i entwickelt. Diese verhält sich leider in einigen Punkten nicht ganz kompatibel zu den offiziellen, d. h. vom MIT und PHILIP ZIMMERMANN veröffentlichten, Versionen, ist aber von PHILIP ZIMMERMANN anerkannt, hat also in gewisser Weise auch einen „offiziellen“ Status. Auf Grundlage dieser Version hat LUTZ DONNERHACKE die Version 2.6.3in programmiert, die einige im Datenformat bereits angelegte, aber sonst nie eingesetzte Besonderheiten verwenden kann, beispielsweise Schlüssel mit Ablaufdatum und Schlüssel, die nur zum Signieren oder nur zum Verschlüsseln verwendet werden sollen.

Heutzutage sind die Versionen 2.6.2, 2.6.2i und 2.6.3i verbreitet.

A.2. Die Versionen 5.x und 6.x

Im Mai 1996 hat PHILIP ZIMMERMANN mit JONATHAN SEYBOLD, DAN LYNCH und anderen für die Weiterentwicklung PGPs die Firma PGP Inc. gegründet. Das erste von dieser Firma veröffentlichte Programm war die Windows-Version von PGP 5.0. Dieser Schritt löste kontroverse Diskussionen aus. Einerseits hatte durch die starke Systemabhängigkeit in den Augen vieler PGP-Nutzenden das Programm stark verloren, auf der anderen Seite war die neue Windows-Oberfläche für viele Neueinsteiger ein starkes Argument, PGP zu verwenden. Einige Zeit später erschienen auch Versionen für die Verwendung unter den gebräuchlichsten Unix-Versionen (ohne graphische Oberfläche), in der Testphase bedauerlicherweise ohne den verwendeten Quelltext. Daß PGP ab Version 5.0 zu einem kommerziellen Produkt geworden war, war keine große Neuerung, denn kommerzielle Versionen gab es bereits vorher und PGP Inc. vertreibt auch Freeware-Versionen der Software, die sich (unter moderaten Einschränkungen, die sich hauptsächlich aus den bereits mehrfach genannten Patent- und Lizenzproblemen ergeben) gratis verwenden lassen. Die Verwendung weiterer Algorithmen (3DES, ElGamal,[⊤] DSS/DSA) weckte kaum Widerspruch, die Ankündigung, RSA und IDEA in absehbarer Zeit fallenlassen zu wollen, wurde hingegen mit wenig Begeisterung aufgenommen, ebenso wie die Änderung des Datenformats – aber mit den neuen Algorithmen erzeugte Nachrichten lassen sich von PGP-Versionen 2.x ohnehin nicht lesen. Insbesondere aber die Lizenzbestimmungen der Freeware-Version stießen (und stoßen immer noch) auf Unverständnis und Ablehnung: Es ist zwar explizit gestattet, den Quelltext des Programms zu vertreiben (Sie finden ihn auf der beiliegenden CD im Verzeichnis `5.0i` in den Verzeichnissen der jeweiligen Betriebssysteme) und für den Hausgebrauch zu modifizieren, aber es dürfen weder geänderte Quelltexte noch die Änderungen selbst (z. B. in Form von „Patches“) veröffentlicht werden. Das ausgesprochen schlechte Handbuch, in dem kaum auf Hintergründe eingegangen wird (und das Wenige ist in großen Teilen falsch), tat ein übriges, um die Begeisterungsstürme in Grenzen zu halten.

Um der in Abschnitt 6.3 angesprochenen Exportproblematik zu begegnen, veröffentlichte PGP Inc. den gesamten Quelltext[•] zusätzlich in

⊤ in der Programmdokumentation fälschlicherweise Diffie-Hellman genannt, vgl. Fußnote[×] auf Seite 272

• Genauer gesagt: Den gesamten Quelltext der Beta-Version 5.0b8

gedruckter Form, denn als Buch konnte das Programm unter Berufung auf „free speech“ exportiert werden. Der Quelltext wurde eingescannt, per OCR (elektronischer Texterkennung) in maschinenlesbare Form übertragen und von etlichen Freiwilligen auf der HIP mit Hilfe der mit abgedruckten Prüfsummen korrekturgelesen. Der hierbei entstandene Quelltext ist als 5.0i bekannt.

Im Oktober 1997 veröffentlichte die Firma PGP Inc. die Version 5.5. Hier war nun die Überraschung groß: Auf einmal enthielt PGP ein „feature“, das je nach Kontext verschiedene Namen hat, u. a. ARR (additional recipient request), CAK (company access to keys) oder GAK (government access to keys). Gemeint ist immer dasselbe: Ein PGP-Schlüssel, der einen Eintrag mit der Bedeutung „an mich verschlüsselte Nachrichten sollen immer auch an jenen Schlüssel verschlüsselt werden“. Um dies zu verstehen, muß man wissen, daß in den USA der Datenschutz bei weitem nicht so entwickelt ist wie in Deutschland, insbesondere wird es dort als vollkommen normal angesehen, daß ein Arbeitgeber den Telephon-, Fax- und Briefverkehr seiner Angestellten am Arbeitsplatz, ihr Internet-Benutzungsverhalten und eben auch ihre E-Mail überwacht. Die Tendenz in Deutschland geht bedauerlicherweise in eine sehr ähnliche Richtung. Um das ganze System wirksam einsetzen zu können, hat PGP Inc. eine weitere Funktion$^{\triangledown}$ eingebaut, mit welcher der PGP installierende Systemadministrator einstellen kann, daß auch sämtliche ausgehende Mail an einen designierten Schlüssel mitverschlüsselt wird und hat einen MTA$^{\spadesuit}$ geschrieben, der eingehende Mail, die nicht an den eingestellten Firmenschlüssel mitverschlüsselt wurde, ablehnt. Besonders überraschend war es für viele Leute, daß bereits die Version 5.0 Code enthält, um ARR-Schlüssel zu unterstützen, auch wenn diese Version noch keine Schlüssel mit dem ARR-Eintrag erzeugen kann. Auf der positiven Seite ist zu vermerken, daß die Version 5.5 wiederum dazu in die Lage versetzt wurde, RSA-Schlüssel zu erzeugen.

Ende des Jahres 1997 wurde die Firma PGP Inc. von der Firma Network Associates (ehemals McAffee) aufgekauft. Die 1998 erschienene Version 6.0 bot Neuerungen hauptsächlich in der Bedienungsoberfläche, beispielsweise die Möglichkeit, den Textinhalt des aktuellen Fensters zu ver- oder entschlüsseln, ohne selbst die Zwischenablage zu bemühen. Das Datenformat wurde ein weiteres mal geändert, an neuen

$\triangledown$ in der Business-Version

$\spadesuit$ MTA: mail transport agent, ein Programm, das sich um die Annahme und Verteilung von Mail kümmert, wie z. B. `sendmail`

Möglichkeiten wurden hier die Option eingeführt, an einen Schlüssel ein Photo anzuhängen (unserer Meinung nach ein „Spielzeug“ ohne praktischen Nutzen) sowie die Markierungen „trusted introducer“ und „trusted metaintroducer“, deren Bedeutung Sie auf Seite 181 erfahren. Außerdem haben Sie im neuen Datenformat die Möglichkeit, in Ihrem Schlüssel weitere Personen (genauer gesagt: Schlüssel) anzugeben, die diesen Schlüssel zurückrufen, also für ungültig erklären dürfen. In der Businessversion und der „personal edition“, nicht aber der Freeware-Version ist des weiteren `PGPDisk` hinzugekommen, das Ihnen gestattet, Partitionen Ihrer Festplatte zu verschlüsseln. Die Unterstützung von RSA und IDEA ist in der Freeware-Version komplett entfernt worden (womit keine Kommunikation mehr mit PGP-Versionen vor der Kommerzialisierung möglich ist, aber die Verwendung von RSA und IDEA in einem kommerziell eingesetzten Produkt kostet Geld, unabhängig davon, ob die entsprechenden Code-Teile verwendet werden, und die Firmen-Policy spricht sich gegen nicht-freie Algorithmen aus). In der Schlüsselverwaltung hat sich auch etwas getan; die bereits vorher verwendeten Unterschlüssel lassen sich jetzt getrennt bearbeiten. Lobend sollte erwähnt werden, daß die mitgelieferte Dokumentation bei PGP 6.0 gegenüber der Dokumentation zu 5.0 wesentliche Verbesserungen erfahren hat. Ein nennenswerter Teil dieser Verbesserungen sind (auf aktuellen Stand gebarchte, aber immer noch deutlich erkennbare) Ausschnitte aus der Dokumentation zu Version 2.3a. Der Abschnitt

> Frühere Versionen von PGP erlaubten es Ihnen, eine ältere Technologie, die RSA genannt wurde, zu verwenden, um Schlüssel zu erzeugen. Mit PGP 5.0 und späteren Versionen haben Sie die Option, eine neue Art von Schlüssel zu erzeugen, die auf der besseren ElGamal-Variante der Diffie-Hellman-Technologie basiert.

ist in dieser Form reines Blendwerk. ElGamal ist von sich aus nicht besser als RSA (auch wenn das zugrundeliegende Problem von Experten als schwieriger zu umgehen angesehen wird); der größte Vorteil ElGamals (insbesondere für die Firma PGP Inc.) ist, daß die Verwendung RSAs Geld kostet.

Kurz vor Drucklegung dieses Buches ist die Version 6.5.1 erschienen. Sie finden in diesem Buch an einigen Stellen Hinweise auf diese Version; für ein umfassende Einarbeitung blieb leider keine Zeit mehr. Die Version 6.5.1 brachte einige Neuerungen mit sich. Die Her-

vorstechendste neue Funktionalität ist mit Sicherheit eine Windows-Systemerweiterung, die IPsec bietet, also eine Verschlüsselung sämtlicher Internet-Kommunikation mit einem Standard, der u. a. auch von Cisco Routern und (mit FreeS/WAN) Linux unterstützt wird. Außerdem ist es mit nunmehr möglich, selbstentschlüsselnde Dateien anzulegen, also (unter Windows) ausführbare Dateien, die beim Starten ein Paßwort verlangen und bei Eingabe des korrekten Paßwortes entschlüsselt werden – ohne, daß auf dem betreffenden Rechner PGP installiert sein muß. Die bereits in PGP 6.0 eingeführte Funktionalität, freie Bereiche der Festplatte zu überschreiben, läßt sich mit Version 6.5 in regelmäßigen Abständen automatisiert aufrufen; diese Option ist sicherlich nur etwas für Anwenderinnen, die genau wissen, was sie mit „Löschen“ meinen. Funktionen wie „Verschlüsseln des aktuellen Fensters“ und ähnliche lassen sich per Tastenkombination (sog. Hotkeys) aufrufen, was die Bedienung stark erleichtert. Optional können Sie sich den Fingerprint eines Schlüssels statt als Kette von Hexadezimalzahlen auch durch eine Liste von Worten darstellen lassen – das kann den Vergleich am Telephon vereinfachen. In dieser Version wird RSA wieder komplett unterstützt und es gibt sogar wieder eine Kommandozeilenversion (allerdings nicht für MS-DOS). Leider konnten wir die Quelltexte für die Unix-Version 6.5.1i erst nach Änderungen überhaupt kompilieren und auch dann ließ sich nur eine Debug-Version erzeugen. Die Bedienung der Kommandozeile wurde im Gegensatz zu Version 5.0 stark an 2.6.x angeglichen.

Detaillierte Beschreibungen der Unterschiede zwischen den einzelnen Windows-Versionen finden Sie in Teil III dieses Buches.

Die neuen Datenformate sind inzwischen (mit leichten Änderungen) unter dem Namen OpenPGP als RfC 2440 veröffentlicht worden.

A.3. GnuPG

Bereits während der Diskussions- und Entstehungsphase des RfC 2440 (OpenPGP) entwickelte WERNER KOCH (mit Unterstützung von MATTHEW SKALA, MICHAEL ROTH und NIKLAS HERNAEUS) eine Implementation, die unter der GNU General Public License veröffentlicht wurde. Dieses Programm nennt sich GnuPG, Sie finden es auf der CD zum Buch im Verzeichnis `GnuPG`. GnuPG ist kompatibel zu PGP 5.0 und 6.0 mit den folgenden Einschränkungen:

- GnuPG kann Schlüssel und Unterschriften erzeugen, die von PGP 5.0 nicht gelesen werden können. Insbesondere Unterschriften

unter langen Dateien und Unterschriften mehrerer Personen unter demselben Dokument können zu Fehlern führen. Das Problem liegt auf Seiten von PGP 5.0.

- PGP 5.x/6.x verwenden ElGamal-Schlüssel ausschließlich zum Verschlüsseln, mit GnuPG (oder einem anderen OpenPGP-kompatiblen Programm) erzeugte ElGamal-Signaturschlüssel werden von PGP 5.x/6.x nicht akzeptiert.
- Ohne zusätzliche Module kann GnuPG weder RSA noch IDEA als Algorithmen verwenden. Derartige Module existieren momentan noch nicht.

GnuPG ist auf Linux und FreeBSD entwickelt worden und läuft auf fast allen Unix-Systemen. Eine Portierung auf Windows existiert, hinkt aber hinter der Entwicklung auf den genannten Unix-Varianten hinterher; eine graphische Benutzeroberfläche für Windows existiert unseres Wissens noch nicht.

B. Kompatibilität der Versionen von PGP

	2.6x	5.0i	5.5.3i	6.0i	6.5.1i	GnuPG
2.6x	+	+⊖	+△	-	-	-
5.0i	+	+	+	+♣	+♣	+
5.5.3i	+	+	+	+	+	+
6.0i	-	+	+	+	+	+
6.5.1i	-	+	+	+	+	+
GnuPG	-	+	+	+	+	+

Es gab einmal eine Version 1.0 von PGP. Vergessen Sie diese, keine spätere Version ist zu ihr kompatibel, sie hat unbrauchbare Algorithmen verwendet und sollte nicht eingesetzt werden.

Version 2.0 ist weitgehend kompatibel mit neueren Versionen, bis hin zu Version 5.0, 6.0 und 6.5. Weil neue Versionen von PGP auch neue Möglichkeiten bieten, können die älteren Versionen manche Dateien, die mit neueren Versionen erzeugt wurden, nicht in jedem Fall bearbeiten. Die Menschen im Entwicklerteam haben sich Mühe gegeben, die internen Datenstrukturen dieser Version von PGP so zu entwerfen, daß sie an künftige Änderungen angepaßt werden können, so daß hoffentlich niemand noch einmal bei einer kommenden Version von PGP die alten Schlüssel wegwerfen und neue generieren muß.

Versionen ab 2.6 erzeugen teilweise Daten, die von früheren PGP-Versionen nicht gelesen werden können. Grund hierfür ist zum Einen die Möglichkeit, Schlüssel mit mehr als 1024 Bit Länge zu verwenden, zum anderen die Tatsache, daß Philip Zimmermann sich mit dem MIT geeinigt hat, die neueren Versionen so zu gestalten, damit mehr Anwen-

⊖ Bei Verwendung von RSA-Schlüsseln. PGP 5.0i kann keine RSA-Schlüssel erzeugen, eingelesene RSA-Schlüssel aber verwenden.
△ Bei Verwendung von RSA-Schlüsseln.
♣ Beim Schlüsselexport aus PGP 6.0i/6.5.1i muß explizit das zu PGP 5 kompatible Format (ohne Photos etc.) gewählt werden.

derinnen auf die patentrechtlich unbedenklichen Versionen umsteigen. Weitere Informationen hierzu finden Sie im vorigen Abschnitt: „Die vielen PGP-Versionen“.

Ab Version 5.0 hat sich das Datenformat geändert. Mit weiteren Modifikationen ist das neue Datenformat in RfC 2440 (OpenPGP) festgeschrieben worden. Die Versionen 5.0 und 6.x können Nachrichten mit den Versionen 2.6.x, 5.x und 6.x austauschen, ersteres aber nur, wenn RSA-Schlüssel verwendet werden. Die Versionen 5.5.3(i) kommt nur mit 5.x und 6.x klar, aber auch nicht immer, sondern nur bei Verwendung von DSS/DH$^{\times}$-Schlüsseln. GnuPG kann mit ihnen allen „reden“; ohne Erweiterungen allerdings mit praktisch denselben Einschränkungen wie 5.5.3i, da RSA und IDEA aus patentrechtlichen Gründen nicht implementiert sind.

Falls Sie jetzt noch Probleme zwischen den Plattformen erwarten, kann ich Sie beruhigen: Abgesehen von den teilweise zu Unrecht nicht anerkannten Klartextunterschriften bei der Version 2.6.3i (näheres finden Sie auf Seite 97) sind uns keine Inkompatibilitäten außer den eben genannten bekannt, Versionen mit gleicher Nummer können generell plattformübergreifend miteinander Daten austauschen.

$\times$ PGP Inc. nennt die ElGamal-Schlüssel, wohl aus Marketinggründen, Diffie-Hellman, daher die Abkürzung.

C. Die beiliegende CD

Die dem Buch beiliegende CD ist erst nach Drucklegung des Buches fertiggestellt worden. Leichte Abweichungen von der hier beschriebenen Verzeichnisstruktur sind daher möglich.

Auf der beiliegenden CD finden Sie die im Buch besprochenen Programme und Utilities, des Weiteren noch einige Hilfsprogramme wie `Alladin Expander` und graphische Frontends für die kommandozeilenorientierten PGP-Versionen für X-Windows, Windows 3.1 und Windows 95/98/NT. Bitte beachten Sie, daß es sich bei diesen Programmen teilweise um Shareware handelt, die Sie nur zu Probezwecken verwenden dürfen, ohne sie käuflich zu erwerben. Mit dem Kaufpreis dieses Buches haben Sie *keine* Lizenz- oder Kaufgebühren für irgendwelche Programme entrichtet.

Die CD verwendet lange Dateinamen mit Hilfe von RockRidge und Joliet. Bedauerlicherweise gestattet es die verwendete Software (`mkisofs 1.11.1rel`) nicht, die erzeugten MS-DOS-Dateinamen vorzugeben. Da die meisten Anwenderinnen und Anwender mit Systemen arbeiten, die die langen Dateinamen verarbeiten können, haben wir uns entschieden, auch keine `TRANS.TBL`-Dateien zu erzeugen – kaum ein System wertet sie aus. Wenn Sie unter MS-DOS arbeiten, möchten wir uns für die Unannehmlichkeiten entschuldigen, die diese Namensgebungen Ihnen verursachen.

Sie finden auf der CD zunächst einmal die Verzeichnisse `2.6xi`, `5.0i`, `5.5i`, `6.0i` und `GnuPG`, in denen die Quelltexte und Installationspakete der jeweiligen PGP-Versionen bzw. GnuPGs liegen. Unterhalb dieser Verzeichnisse liegen Verzeichnisse für die jeweils unterstützen Betriebssysteme bzw. für die Quelltexte.

Im Verzeichnis `Doku` finden Sie die in den Danksagungen bereits erwähnte Anleitung zu den Windows-Versionen PGPs, geschrieben von KAI RAVEN. Leider enthält sie ein paar Fehler; wir möchten sie Ihnen dennoch unverändert und unkommentiert zur Verfügung stellen. Außerdem haben wir in dieses Verzeichnis die beiden für PGP relevanten RfCs aufgenommen.

Das Verzeichnis `Frontends` enthält – wiederum aufgeteilt in die verschiedenen Betriebssysteme – graphische Benutzeroberflächen für die Kommandozeilenversionen PGPs. Analog dazu finden Sie im Verzeichnis `Plugins` alles, was Sie benötigen, um die Einbindung PGPs in Ihr Mailprogramm oder Ihren Editor zu vereinfachen. Neben den in Teil III besprochenen Plugins finden Sie hier auch Zusatzprogramme, mit denen die Kommandozeilenversionen eingebunden werden können.

Das Verzeichnis `Remailer` befaßt sich mit Software, die die Verwendung anonymer Remailer (siehe Abschnitt 5.9 auf Seite 44) unterstützen. Außerdem finden Sie dort einige Remailer-Programme.

Im Verzeichnis `Stego` finden Sie Programme, die sich am Bereich der Steganographie versuchen, ein Komplex, den wir in diesem Buch komplett ausgespart haben. Steganographie beschäftigt sich im Gegensatz zur Kryptographie nicht damit, Nachrichten zu verschleiern, sondern damit, sie zu verstecken.

Im Verzeichnis `Utilities` haben wir für Sie Hilfsprogramme wie den Acrobat Reader und Kompressionsprogramme versammelt. Hier finden Sie auch ein Hilfsprogramm, das das Wiederherstellen gelöschter Dateien unter Linux zu vereinfachen sucht und dergleichen mehr. Die mehrfach genannte Freeware `Alladin Expander` finden Sie auch hier, unter `Utilities/Windows/Packer/alex50.exe`.

Schließlich bleibt noch das Verzeichnis `cfs` zu nennen. `Cfs` steht als Abkürzung für „cryptographic file system“; in diesem Verzeichnis finden Sie Programme und Betriebssystemerweiterungen, die es Ihnen erlauben, Ihre Festplatte oder Teile davon zu verschlüsseln.

D. Kurz vorgestellt: Die Verschlüsselungsalgorithmen

D.1. IDEA

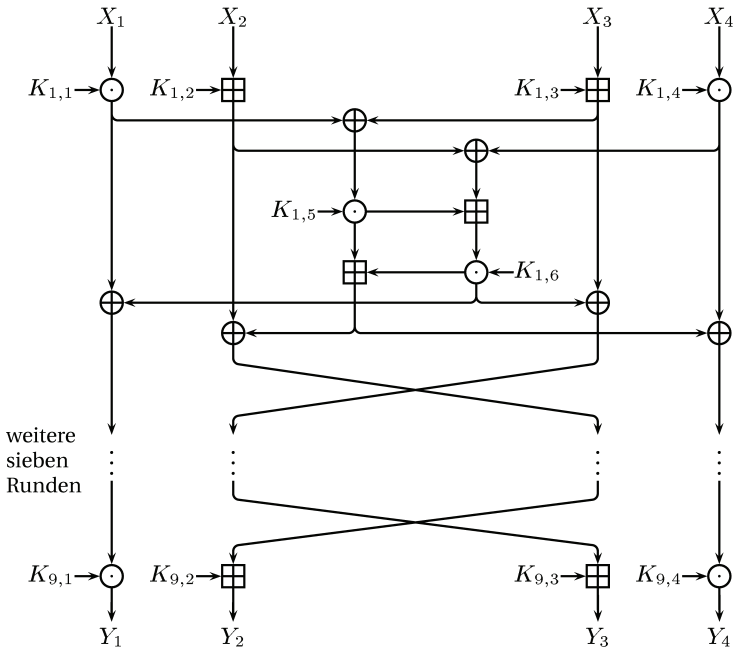
IDEA™ soll hier stellvertretend für die symmetrischen Verschlüsselungsalgorithmen behandelt werden. Weitere Algorithmen finden Sie in [Sti95, Sch95]. IDEA basiert auf der Kombination einfacher Rechenoperationen. Verwendet werden:

1. Bitweise Addition zweier Zahlen ohne Übertrag (XOR)
2. Addition zweier Zahlen ohne Berücksichtigung des Übertrags über 2^{16} hinaus
3. Multiplikation zweier Zahlen und Bildung des Restes nach Division durch $2^{16}+1$. Hierbei werden 0 und 2^{16} besonders behandelt: Vor Beginn der Multiplikation wird eine 0 durch 2^{16} ersetzt,♢ das Ergebnis 2^{16} wiederum wird als 0 interpretiert. Daraus folgt: 0„mal“$0=1$.

Diese drei Operationen wurden nicht absolut willkürlich gewählt, sondern aufgrund einer ganz speziellen mathematischen Eigenschaft: Sie bilden zwei 16-Bit-Zahlen a,b auf eine 16-Bit-Zahl c ab und sind sowohl in b invertierbar. Das heißt: Zu jedem b gibt es ein b', so daß $(a\cdot b)\cdot b' = a$ für alle a gilt. Das ist die entscheidende Eigenschaft, die das Entschlüsseln (bei bekanntem Schlüssel) erlaubt. Nebenbei sind die Operationen auch in a invertierbar (und überhaupt kommutativ), was nahelegt, daß ihr Ausgabewert wünschenswert stark vom Schlüssel abhängt.

Aus dem 128-Bit-Schlüssel werden Teilschlüssel berechnet, und zwar $S_{1,1}$ bis $S_{8,6}$ und $S_{9,1}$ bis $S_{9,4}$. Hierfür wird der Schlüssel in acht 16 Bit große Teile geteilt, diese ergeben $S_{1,1}$ bis $S_{1,6}$, $S_{2,1}$ und $S_{2,2}$. Anschließend

♢ Vor Beginn der Multiplikation kann keiner der Werte 2^{16} sein, da nur 16-Bit-Zahlen verwendet werden.



X_i	Klartext-Teilblock (16 Bit)
Y_i	Chiffrat-Teilblock (16 Bit)
$K_{r,i}$	Teilschlüssel i in Runde r
$\oplus$	Bitweises exclusiv-oder
$\boxplus$	Addition modulo 2^{16}
$\odot$	Multiplikation modulo $2^{16}+1$, wobei die 0 als 2^{16} angesehen wird

Abbildung D.1: IDEA™

wird der gesamte 128-Bit-Schlüssel um 25 Bit nach links rotiert und wieder in acht Blöcke zu je 16 Bit unterteilt, die dann $S_{2,3}$ bis $S_{2,6}$ und $S_{3,1}$ bis $S_{3,4}$ werden. Dann wird wieder rotiert und so weiter.

Die eigentliche Verschlüsselung läuft so ab, daß ein Klartextblock von 64 Bit Länge in vier Blöcke zu je 16 Bit eingeteilt wird, die anschließend dem Verfahren in Abb. D.1 unterworfen werden. Dargestellt ist nur der erste Durchlauf, das Verfahren wird achtmal angewendet.

Zum Entschlüsseln kann dasselbe Verfahren verwendet werden. Damit dabei der ursprüngliche Klartext erhalten bleibt, müssen leicht andere Teilschlüssel verwendet werden.

D.2. RSA

RSA, das wohl berühmteste asymmetrische Verschlüsselungsverfahren, ist benannt worden nach RIVEST, SHAMIR und ADLEMAN, seinen Entwicklern. Als sie es 1977 veröffentlichten, war es das einzige öffentlich bekannte Verfahren, daß die 1976 von DIFFIE und HELLMAN publizierte Idee öffentlicher Schlüssel tatsächlich einsetzen konnte.

Das System basiert auf Rechnungen im Ring der ganzen Zahlen modulo pq, wobei p und q zwei verschiedene Primzahlen sind. In diesem System zu rechnen, geht ebenso vor sich wie gewohnt, nur, daß vom Ergebnis nur der Rest bei Division durch pq behalten wird. Als Zeichen, daß wir diese Art der Rechnung durchführen, verwenden wir das $\equiv$ anstelle des üblichen $=$ und schreiben den „Modulus“, also das pq, hinter die Rechnung. Wenn wir als Beispiel $pq = 15$ setzen, dann sind folgende Rechnungen korrekt:

$$\begin{aligned} 2 + 5 &\equiv 7 \pmod{15} \\ 2 * 5 &\equiv 10 \pmod{15} \\ 4 * 5 &\equiv 5 \pmod{15} \\ 4 * 4 &\equiv 1 \pmod{15} \\ \Rightarrow 1 \,/\, 4 &\equiv 4 \pmod{15}, \text{ denn } x = 1/y \text{ bedeutet } x * y = 1. \end{aligned}$$

Von besonderem Interesse sind hier die Exponentialfunktionen:

$$\begin{aligned} 5^2 &\equiv 10 \pmod{15} \\ 4^7 &\equiv 4 \pmod{15}, \end{aligned}$$

denn es ist kein effizientes Verfahren bekannt, diese Rechnung umzukehren, d. h. es ist keine Möglichkeit bekannt, in annehmbarer Zeit Probleme wie $x^5 \equiv 12$ zu lösen. (Auf der Schwierigkeit des diskreten Logarithmus, also der Lösung von $6^x \equiv 8$ etc., beruhen andere Verfahren, beispielsweise ElGamal.) Weiterhin interessant ist eine Beziehung, die schon Euler bekannt war:

$$a^{\phi(x)} \equiv 1 \pmod{x}$$

Hier sind a und x beliebig, davon abgesehen, daß sie keinen gemeinsamen Teiler größer als 1 haben. $\equiv$ ist das Zeichen für das oben erwähnte Modulo-Rechnen, und zwar modulo x. Ein weiteres neues Zeichen steht in dieser Formel: $\phi(x)$ ist die „Eulersche Phi-Funktion“. Für uns wichtig ist nur, daß $\phi(pq) = (p-1)(q-1)$ gilt, wiederum für die Primzahlen p und q. Eine kurze Rechnung ergibt für eine beliebige ganze Zahl k:

$$\begin{aligned} a^{\phi(pq)} &\equiv 1 \pmod{pq} \\ a^{(k*\phi(pq))} &\equiv 1 \pmod{pq} \\ a^{(k*\phi(pq)+1)} &\equiv a \pmod{pq} \end{aligned}$$

Wenn wir nun zwei ganze Zahlen d und e einführen, von denen wir verlangen, daß $de = k * \phi(pq) + 1$ gelten soll,[⊗] dann erhalten wir:

$$a^{de} \equiv a \pmod{pq}$$

$$\begin{aligned} a^d &\equiv b \pmod{pq} \\ b^e &\equiv a \pmod{pq} \end{aligned}$$

Wobei die Kenntnis von b und d nicht ausreicht, um a zu berechnen. RSA funktioniert nun so, daß als öffentlicher Schlüssel d und das Produkt pq veröffentlicht werden und die Nachrichten a damit wie eben beschrieben verschlüsselt werden. Die verschlüsselten Nachrichten (b) können dann bedenkenlos versandt werden, da sie ohne e nicht entschlüsselt werden können.[⊙]

⊗ Es gibt zu jedem d, das keinen gemeinsamen Teiler mit $\phi(pq)$ hat, so ein e. Das zu beweisen, würde hier aber zu weit führen.

⊙ Zumindest ist kein Verfahren öffentlich bekannt . . .

Angriffsmöglichkeiten auf RSA

Aus a und b e berechnen Dies nennt sich in der Sprache der Mathematiker „diskreter Logarithmus“ und gilt derzeit als praktisch (d. h. in „kurzer“ Zeit) unlösbar.

Aus pq und d das geheime e berechnen Hierzu muß nach derzeitigem Wissensstand pq in seine Faktoren zerlegt werden, was ebenfalls für nicht in sinnvoller Zeit lösbar gehalten wird. Als Anhaltspunkt: RON RIVEST hat 1995 geschätzt, daß es im Jahre 2010 möglich sein dürfte, eine 1689-Bit-Zahl zu faktorisieren – zu Kosten von 25 Milliarden Euro. Bei einer Investitionssumme von 25 Millionen Euro schätzte er die Grenze auf 705 Bit.

Ein spezieller Angriff auf Nachrichten mit mehreren Empfängern Wenn dasselbe a an mehrere Empfänger verschlüsselt wird, wobei dasselbe d verwendet wird,[∨] ist es mit einem kleinen mathematischen Kunstgriff möglich, die Nachricht zu entschlüsseln. Dies betrifft PGP nicht, da hier beim Verschlüsseln an mehrere Empfänger der IDEA-Schlüssel stets mit Zufallszahlen aufgefüllt wird, die für jeden Empfänger verschieden sind.

D.3. ElGamal

Die Sicherheit ElGamals beruht auf der Schwierigkeit, diskrete Logarithmen zu finden, also der Schwierigkeit, eine Gleichung der Form

$$y \equiv g^x \pmod{n}$$

zu gegebenen y, g und p nach x aufzulösen. (Zur Notation siehe Abschnitt D.2.) Dieses Problem ist in vielen Spezialfällen (d. h. für n einer bestimmten allgemeinen Form) gelöst, der allgemeine Fall ist aber nach wie vor ein hartes Problem. Übrigens würde ein erfolgreicher Angriff auf das Problem des diskreten Logarithmus auch die Sicherheit des RSA-Verfahrens zunichte machen, da jeder den privaten Schlüssel ausrechnen könnte.

Um einen ElGamal-Schlüssel zu berechnen, wählt man zufällig eine Primzahl p (diese Zahl sollte recht groß sein, z. B. 1024 Bit) und zwei Zah-

[∨] Das ist nicht unwahrscheinlich. PGP versucht standardmäßig immer, die 17 zu verwenden.

len g und x, die kleiner als $p-1$ sind. Dann berechnet man

$$y = g^x \pmod{p}.$$

Der öffentliche Schlüssel ist (y, g, p), der private Schlüssel (x, g, p). g und p können ohne Probleme für viele Teilnehmer identisch sein. $p-1$ ist immer gerade, $(p-1)/2$ sollte keine kleinen Faktoren haben, weil sonst ein Algorithmus von POHLIG und HELLMAN verwendet werden kann, um den diskreten Logarithmus etwas schneller zu berechnen. PGP wählt für jeden Benutzer ein eigenes p; um diese Bedingung an $(p-1)/2$ zu erfüllen, wählt es dazu Primzahlen $p_1, \ldots, p_n$ und probiert, ob $p = 1+2\prod_{i=1}^{n} p_i$ eine Primzahl ist. Wenn nicht, werden andere $p_1, \ldots, p_{n'}$ gewählt.

Unterschriften

ElGamal-Unterschriften werden wie folgt erzeugt: Zunächst wähle ein zufälliges k, das keinen gemeinsamen Teiler mit $p-1$ hat und berechne h, den Hash-Wert der Nachricht. Dann berechne man a und b, so daß

$$\begin{aligned} a &\equiv g^k \pmod{p} \\ h &\equiv xa + bk \pmod{p-1}. \end{aligned}$$

b läßt sich mit Hilfe des erweiterten euklidischen Algorithmus berechnen, wie, ist hier nicht weiter wichtig. Das Paar (a, b) ist die Unterschrift. Sie läßt sich prüfen, indem die folgende Gleichung auf Korrektheit getestet wird:

$$y^a a^b \equiv g^h \pmod{p}$$

Das funktioniert, weil bei einer korrekten Unterschrift gilt:

$$\begin{aligned} y^a a^b &\equiv (g^x)^a \left(g^k\right)^b \\ &\equiv g^{xa} g^{kb} \\ &\equiv g^{xa+kb} \\ &\equiv g^h \pmod{p} \end{aligned}$$

Verschlüsselung

Eine Nachricht M mit ElGamal zu verschlüsseln, funktioniert so: Zunächst wähle man wiederum ein zufälliges k ohne gemeinsame Teiler mit $p-1$. Dann berechne man

$$\begin{aligned} a &\equiv g^k \pmod{p} \\ b &\equiv y^k M \pmod{p}. \end{aligned}$$

Das Paar (a,b) ist die verschlüsselte Nachricht. Man beachte, daß die Länge der Nachricht hierbei mindestens verdoppelt wird. Zum Dekodieren braucht der Empfänger nur

$$\frac{b}{a^x} \equiv \frac{y^k M}{(g^k)^x} \equiv \frac{y^k M}{y^k} \equiv M \pmod{p}$$

zu berechnen.

E. Rechtsfragen

E.1. Warenzeichen, Copyright, Garantie

„Pretty Good Privacy“ ist eingetragenes Warenzeichen in mehreren Ländern von Philip Zimmermann und von Networks Associates, Inc. „und angegliederten Unternehmen“. PGP ist © Copyright by Philip R. Zimmermann 1990-1993 und © Copyright 1990-1999 Network Associates. Philip Zimmermann beansprucht auch das Copyright für das PGP-Handbuch und für Übersetzungen von Handbuch und Software in andere Sprachen. Nach dem deutschen Urheberrecht liegen die Rechte für diese Übersetzung bei uns, da wir sie geschrieben haben. Weitere Handelsnamen und Markennamen werden in diesem Handbuch ohne nähere Kennzeichnung verwendet.

Wir übernehmen keine Haftung für Schäden, die aus der Nutzung der Software entstehen, auch dann nicht, wenn die Schäden aus Fehlern des Handbuchs oder der Software resultieren. Wir machen keine rechtsverbindlichen Angaben über die Verkaufbarkeit der Software oder ihre Brauchbarkeit für bestimmte Anwendungen. Die Software wird nur in der bestehenden Form zur Verfügung gestellt, ohne jede explizite oder implizite Garantie.

Vervielfältigungen, Änderungen oder Übertragungen in andere Formen, auch auszugsweise, sind nur nach schriftlicher Erlaubnis der Autoren oder des Verlages gestattet. Dies gilt auch für evtl. zur Verfügung gestellte elektronische Versionen.

E.2. Patentrechte auf die Algorithmen

Das Verschlüsselungsverfahren RSA wurde am MIT entwickelt. Dem MIT wurde ein Patent auf RSA erteilt (U. S. Patent #4,405,829, erteilt am 20. September 1983). Eine kalifornische Firma namens Public Key Partners (PKP) besitzt die alleinigen Rechte an diesem Patent für den Verkauf und die Lizensierung des RSA-Verschlüsselungsverfahrens.

Für Anwender außerhalb der USA sei angemerkt, daß das US-Patent auf RSA nur innerhalb der USA gilt und daß es kein RSA-Patent in anderen Ländern gibt.

US-Bundesbehörden können RSA nutzen, weil die Entwicklung von RSA staatlich durch Zuschüsse der National Science Foundation und der US-Navy finanziert wurde. Die Verwendung von PGP durch Stellen der US-Regierung unterliegt jedoch Einschränkungen, die sich aus der Einigung Philip Zimmermanns mit ViaCrypt ergeben. Hierzu später mehr.

PKP erhielt nicht nur die ausschließlichen Patentrechte für RSA, sondern auch die ausschließlichen Rechte für drei andere Patente für asymmetrische Verschlüsselungsverfahren, die an der Stanford University mit Bundeszuschüssen entwickelt wurden. Im Prinzip entscheidet damit in den USA eine einzige Firma über die Verwendung von public key Verschlüsselungssystemen. PKP beansprucht sogar das Patentrecht an dem grundlegenden Konzept der Kryptographie mit öffentlichen Schlüsseln, unabhängig davon, wie intelligent auch immer ein neuer Algorithmus sein mag, der unabhängig von PKP entwickelt werden könnte.

Wir halten ein derart umfassendes Monopol für gefährlich, weil wir der Meinung sind, daß Kryptographie und insbesondere Kryptographie mit öffentlichen Schlüsseln einen zentralen Beitrag zum Schutz der Bürgerrechte und der Privatsphäre in unserer immer mehr verkabelten Gesellschaft leisten kann. Zumindest setzt das Monopol von PKP diese lebenswichtige Technologie dem Risiko der Einflußnahme durch Regierungen und mächtige Firmen aus. Das Patent auf Diffie-Hellman ist am 29. April 1997 ausgelaufen, damit sollte auch dem Argument, das grundlegende Konzept der asymmetrischen Verschlüsselung sei patentiert, jegliche Grundlage entzogen sein. ElGamal war niemals patentiert, die Firma PKP stand jedoch auf dem Standpunkt, ElGamal falle unter das Diffie-Hellman-Patent [Sch95]. Das RSA-Patent wird am 20. September 2000 auslaufen. Ein weiteres Patent, für einen Algorithmus namens „knapsack algorithm“ auch „Merkle-Hellman“ genannt, ist in den USA am 19. August 1997 ausgelaufen. Dieser Algorithmus ist 1983 geknackt worden, inzwischen auch sämtliche Variationen. Daher verwendet den Algorithmus ohnehin niemand mehr, folgerichtig sind auch die entsprechenden Patente auf Variationen in anderen Ländern uninteressant.

Seit der Version 2.5 (vertrieben vom MIT, dem Inhaber des originalen Patentes auf RSA) verwendet die Freeware-Version von PGP die RSAREF-Routinen, die innerhalb der USA für nicht-kommerzielle Anwendungen benutzt werden dürfen. Seit Version 5.0 wird ElGamal als asymmetri-

scher Verschlüsselungsalgorithmus und DSA/DSS (eine weitere Variante derselben Mathematik wie in Diffie-Hellman und ElGamal) als Algorithmus für die Unterschriften verwendet. Gerüchten zufolge beansprucht der deutsche Mathematiker SCHNORR für sich, daß der DSA eine Patentrechtsverletzung seines Algorithmus darstellt. Die weltweiten Rechte am entsprechenden Patent liegen laut [Sch95] seit 1993 bei PKP, das US-Patent läuft am 19. Februar 2008 aus. Die weiteren Patente, gegen die der DSA nach Meinung der jeweiligen Rechte-Inhaber verstößt (Diffie-Hellman und Merkle-Hellman), sind, wie eben erwähnt, inzwischen ausgelaufen. Die NSA (genauer: David Kratz) hat ein US-Patent auf den DSA erhalten und das NIST (das „National Institute of Standards and Technology“, die Behörde die den DSS wie auch ehemals den DES und in naher Zukunft den AES veröffentlicht und zum Standard erklärt hat) hat veröffentlicht, die Technologie sei weltweit frei einsetzbar. (Wozu sie dann ein Patent beantragt haben, ist mir unklar, aber ich bin auch kein Anwalt.)

Die PGP Version 2.0 entstand aus der gemeinsamen Arbeit eines internationalen Programmierteams, das unter PHILIP ZIMMERMANNS Leitung Verbesserungen gegenüber der ersten Version implementierte. Diese Version wurde von BRANKO LANCASTER in den Niederlanden und von PETER GUTMANN in Neuseeland veröffentlicht, außerhalb der Reichweite des Patentgesetzes der USA. Obwohl es nur in Europa und Neuseeland veröffentlicht wurde, verbreitete sich PGP spontan in die USA, ohne daß das Entwicklungsteam etwas damit zu tun gehabt hätte.

Das blockorientierte Verschlüsselungsverfahren IDEA, das die meisten PGP-Varianten verwenden (können), unterliegt in Europa und den USA einem Patent,⊕ das der ETH Zürich und der Schweizer Firma Ascom Systec AG gehört. Die schweizerische Patentnummer ist PCT/CH91/00117. Die US-Patentnummer ist US005214703, die europäische Patentnummer lautet EP 0 482 154 B1. Für die nicht-kommerzielle Verwendung von IDEA werden keine Lizenzgebühren verlangt. Staatliche und kommerzielle Anwenderinnen benötigen eine lizensierte PGP-Version von Network Associates, die Kosten schließen eine Lizenz für IDEA und RSA mit ein.

Die bei PGP verwendeten ZIP-Kompressionsroutinen stammen aus Freeware-Quellcode und werden mit Erlaubnis des Autors verwendet.

⊕ Es ist richtig, daß nach deutschem Recht ein Algorithmus nicht patentierbar ist; die Anwendung eines speziellen Algorithmus für einen bestimmten Zweck, beispielsweise die Verschlüsselung einer Nachricht, scheint aber patentierbar zu sein. Im Zweifelsfall wenden Sie sich bitte an einen Patentanwalt Ihres Vertrauens.

Uns sind keine Patente bekannt, die für die Kompressionsalgorithmen erteilt worden wären, aber Sie können diese Frage gerne selbst genauer untersuchen.

E.3. Lizensierung und Vertrieb

PGP 2.6 ist in den USA unter den Bedingungen der RSAREF-Lizenz vom Massachusetts Institute of Technology erhältlich. Jeder darf die Freeware-Versionen von PGP frei verbreiten und verwenden, ohne dafür etwas zu bezahlen, vorausgesetzt, dies dient privaten, nicht-kommerziellen Zwecken. Kommerzielle Anwender wenden sich bitte an Network Associates, unter der URL `http://www.pgp.com` werden Sie direkt auf die entsprechenden Webseiten geleitet. Alle Hinweise auf Copyright, Patente und geschützte Handelsnamen müssen bei der Verbreitung PGPs erhalten bleiben, die Dokumentation (im US-englischen Original) muß mitverbreitet werden.

PHILIP ZIMMERMANN mußte im Sommer 1993 ein Abkommen mit ViaCrypt schließen, das dieser Firma die exklusiven Rechte an einer kommerziellen Version von PGP gab, um Unternehmen einen rechtlich abgesicherten Weg zu bieten, PGP zu verwenden, ohne von PKP wegen Patentrechtsverletzungen belangt zu werden. Um PGP auf lange Sicht als Standard etablieren zu können, mußte das juristische Stigma, das mit der Verwendung von RSA verbunden war, beseitigt werden. ViaCrypt besaß bereits eine Lizenz von PKP, Produkte herzustellen, zu benutzen und zu vertreiben, die RSA verwenden. ViaCrypt bot sich als Weg aus der Illegalität an, in der PGP vorher operierte. Sie konnten eine kommerzielle Version von PGP verkaufen, wenn er ihnen die Lizenz dazu geben würde. Um PGP eine Zukunft im kommerziellen Sektor zu bieten, hat Philip Zimmermann diesen Weg eingeschlagen. Das war notwendig, um PGP das Überleben zu sichern.

Im Jahre 1996 schließlich wurde diese Zusammenarbeit beendet und PHILIP ZIMMERMANN gründete gemeinsam mit JOANTHAN SEYBOLD, DAN LYNCH und anderen die Firma PGP Inc., von der alle PGP-Versionen ab 5.0 veröffentlicht wurden. Diese Firma ist Ende 1997 von der Firma Network Associates aufgekauft worden; dies führte zu einigem Aufruhr, da diese Firma$^\infty$ als Mitglied in der „key recovery alliance“ bekannt war,

∞ Genauer: Eine weitere Tochterfirma, die in etwa zur selben Zeit wie PGP Inc. akquiriert wurde

dem Zusammenschluß einiger US-amerikanischer Firmen, die sich als Ziel gesetzt haben, Verschlüsselungssysteme zu etablieren, bei denen staatliche Stellen und Vorgesetzte Zugriff auf die „geheimen“ Schlüssel haben.

PGP ist keine Shareware. Es gibt Freeware-Versionen, veröffentlicht als gesellschaftliche Dienstleistung, und kommerzielle Versionen, ursprünglich aus patentrechtlichen Gründen ins Leben gerufen. Daß PGP (in den Freeware-Versionen) für den Privatgebrauch kostenlos verwendet werden kann und darf, ermutigt viele Menschen, PGP auch zu verwenden. Dies wird hoffentlich größere soziale Auswirkungen haben, woraus sich eine weite Verbreitung starker Verschlüsselung ergeben könnte, was seinerseits wiederum wünschenswerte soziale Auswirkungen hätte.

Scheuen Sie sich nicht, die Freeware-Versionen als vollständiges Paket soweit wie möglich zu verbreiten. Geben Sie es allen Ihren Freunden. Wenn Sie Zugang zu MailBoxen haben, stellen Sie PGP in möglichst allen MailBoxen öffentlich zur Verfügung. Auch den Quellcode können Sie beliebig verbreiten.

Unabhängig von den komplizierten und teilweise überlappenden Beschränkungen und Bedingungen der verschiedenen Patent- und Urheberrechte (mindestens RSA, RSAREF und IDEA), die verschiedene Institutionen haben, gilt eine weitere Beschränkung auf die Benutzung von PGP, die sich aus der oben genannten Einigung mit ViaCrypt und für die aktuellen Versionen aus den Lizenzbestimmungen der Firmen PGP Inc. und Network Associates, Inc. ergibt: Die Freeware-Versionen dürfen nur privat und nicht kommerziell genutzt werden.

PGP darf unter keinen Umständen ohne die Dokumentation verbreitet werden. Das schließt die US-englischsprachige Anleitung und bei den Versionen, die mit RSAREF arbeiten, die RSAREF-Lizenz mit ein.

Es gibt so viele fremdsprachige Übersetzungen von PGP, daß die meisten Sprachkits nicht im Standardpaket von PGP enthalten sind, um Speicherplatz zu sparen. Einzelne Sprachkits können Sie aus einer großen Zahl unabhängiger Quellen beziehen. Häufig sind es die gleichen Quellen, bei denen Sie auch das eigentliche PGP-Paket finden. Diese Kits enthalten übersetzte Versionen von `language.txt`, `pgp.hlp` und manchmal auch Teile des Handbuchs. Wenn Sie ein Sprachkit für eine bestimmte Sprache suchen, probieren Sie es am besten in den entsprechenden Internetgruppen oder auf `www.pgpi.org`.

Wenn Sie Usenet-Anschluß haben, beobachten Sie die Newsgroups `sci.crypt` und die PGP-speziellen Newsgroups `alt.security.pgp` und `z-netz.alt.pgp.allgemein`, um Ankündigungen neuer PGP-Versionen zu erhalten. Als weitere Informationsquelle ist natürlich `www.pgpi.com` und `www.pgpi.org` zu nennen.

Bitte beachten Sie bei obigen Ausführungen, daß es die US-Regierung als illegalen Export betrachtet, wenn Sie von außerhalb der USA PGP oder andere Programme oder Daten, die den Exportbeschränkungen unterliegen von einem US-amerikanischen ftp-Server oder einer US-amerikanischen MailBox kopieren. Möglicherweise gefährden Sie damit sogar einen US-amerikanischen Systembetreiber. Nach einer Meldung in `alt.security.pgp` wurde in den USA schon gegen einen MailBoxbetreiber ermittelt, der PGP öffentlich angeboten hat.

In Deutschland ist zu erwarten, daß neue Versionen von PGP nach kurzer Zeit, in der der Sourcecode auf offensichtliche Manipulationen geprüft wird, in diversen Systemen und auf diversen ftp-Servern, u. a. in der //BIONIC zu finden sein werden. Diese MailBox ist erreichbar unter der Nummer 0521-68 000 (ISDN: 0521-9 68 58 69), Loginname `PGP`, kein Paßwort. Ebenso werden wir diese Versionen auch auf `ftp://ftp.foebud.org/pub/pgp` zur Verfügung stellen.

Bei künftigen Versionen von PGP kann sich unter Umständen das Datenformat von Nachrichten, Unterschriften, Schlüsseln oder Schlüsseldateien ändern, wenn dadurch wichtige neue Funktionen ermöglicht werden. Daraus können sich Probleme mit der Kompatibilität zur gegenwärtig aktuellen Version ergeben. Diese Versionen werden möglicherweise Konvertierungsprogramme für alte Schlüssel enthalten, aber Nachrichten, die mit neuen PGP-Versionen verschlüsselt wurden, werden eventuell nicht kompatibel zu den alten Versionen von PGP sein.

F. Computerorientierte politische Vereinigungen in Deutschland

- Computernetzwerk Linksysteme (/CL-Netz)
  Kommunikation und Neue Medien e. V.
  Postfach 19 05 20
  D-80605 München
  Tel: +49-89-167 51 06
  Fax: +49-89-13 14 06
  eMail: <cl-service@link-m.de>
  Infoklick: http://www.cl-netz.de/
- Chaos Computer Club e. V. (CCC)
  Lokstedter Weg 72
  D-20251 Hamburg
  Tel: +49-40-40 18 01-0
  Fax: +49-40-491 76 89
  eMail: <info@ccc.de>
  Infoklick: http://www.ccc.de/
- DVD – Deutsche Vereinigung für Datenschutz
  Bonner Talweg 33-35
  D-53113 Bonn
  Tel: +49-228-22 24 98
  Infoklick: http://www.aktiv.org/DVD/
- Forum InformatikerInnen für Frieden und gesellschaftliche Verantwortung e. V. (FIFF)
  Reuterstraße 44
  D-53113 Bonn
  Tel: +49-228-21 95 48 (di 9-15 + do 16-19 Uhr)
  Fax: +49-228-21 49 24
  eMail: <fiff@fiff.gun.de>
  Infoklick: http://fiff.gun.de/

- FITUG- Förderverein Informatik und Gesellschaft
  Infoklick: `http://www.fitug.de/`
- FoeBuD e. V.
  Marktstraße 18
  D-33602 Bielefeld
  Tel: +49-521-17 52 54 (mo-fr 17-19 Uhr)
  Fax: +49-521-6 11 72
  eMail: `<foebud@foebud.org>`
  Infoklick: `http://www.foebud.org/`
- GDD – Gesellschaft für Datenschutz und Datensicherung
  Andreas Jaspers
  Irmintrudisstraße 1b
  D-59111 Bonn
  Tel: +49-228-69 49 19
  Fax: +49-228-69 56 98
  Infoklick: `http://www.gdd.de/`
- Humanistische Union (HU) e. V.
  Bräuhausstraße 2
  D-80331 München
  Tel. +49-89-22 64 41
- Teletrust Deutschland e.V.
  Helmut Reimer
  Eichendorffstraße 16
  D-99096 Erfurt
  Tel: +49-361-346 05 31
  Fax: +49-361-345 39 57
  Infoklick: `http://www.teletrust.de/`

Literaturverzeichnis

[Bam93] J. Bamford, *The Puzzle Palace: Inside the National Security Agency, America's Most Secret Intelligence Organization* (Penguin Books/Viking Penguin, 1993), 2. Auflage, ISBN 0-14-023116-1.

[Beu93] A. Beutelspacher, *Kryptologie* (Vieweg, 1993), 3. Auflage, ISBN 3-528-28990-2.

[Den82] D. Denning, *Cryptography and Data Security* (Addison-Wesley, 1982).

[Den83] D. Denning, *Protecting Public Keys and Signature Keys* (IEEE Computer, 1983).

[Gar95] S. Garfinkel, *PGP: Pretty Good Privacy* (O'Reilly & Associates, Inc., 1995), ISBN 1-56592-098-8.

[Hel79] M. E. Hellman, *The Mathematics of Public-Key Cryptography*, *Scientific American*, Aug 1979.

[Lai91] X. Lai, *Markov Ciphers and Differential Cryptoanalysis*, in *Advances in Cryptology—EUROCRYPT'91*, 1991.

[Lai92] X. Lai, *On the Design and Security of Block Ciphers* (Institute for Signal and Information Processing, ETH-Zentrum, Zurich, 1992).

[Luc99a] N. Luckhardt, *Pretty Good Privacy - Teil 1*, *c't*, Band 12, Seiten 212–214, 1999.

[Luc99b] N. Luckhardt, *Pretty Good Privacy - Teil 2*, *c't*, Band 13, Seiten 208–210, 1999.

[Luc99c] N. Luckhardt, *Pretty Good Privacy - Teil 3: Dateibearbeitung und geteilte Schlüssel*, *c't*, Band 16, Seiten 172–175, 1999.

[Riv92] R. Rivest, *The MD5 Message Digest Algorithm*, RFC 1321, MIT Laboratory for Computer Science, April 1992.

[Sch95] B. Schneier, *Applied Cryptography: Protocols, Algorithms and Source Code in C* (John Wiley & Sons, 1995), 2. Auflage.

[SH99a] C. Schulzki-Haddouti, *Grünes Licht für Kryptographie*, *c't*, Band 13, Seiten 46, 1999.

[SH99b] C. Schulzki-Haddouti, *Mehr verschlüsseln! – Interview mit Bundeswirtschaftsminister Dr. Werner Müller*, *c't*, Band 21, Seiten 46–47, 1999.

[Sti95] D. R. Stinson, *Cryptography: Theory and practice* (CRC Press, 1995), ISBN 0-8493-8521-0.

[Wal93] P. Wallich, *Electronic Envelopes*, *Scientific American*, Seite 30 ff., Februar 1993.

[Wal94] P. Wallich, *Piraten im Datennetz*, *Spektrum der Wissenschaft*, Seite 64 ff., Mai 1994. Eine relativ ausführliche Einleitung in die Sicherheitsproblematiken im Internet.

[Zim88] P. Zimmermann, *A proposed Standard Format for RSA Cryptosystems*, in *Advances in Computer Security* (Herausgeber R. Turn), Band III (Artech House, 1988).

G. Glossar

Algorithmus Eine Rechenvorschrift. (Informatikerinnen mögen uns die stark verkürzte Erklärung verzeihen.)

Angriff Der Versuch, verschlüsselte oder sonstige nicht offen zugängliche Daten zu erhalten, d. h. sie ohne Berechtigung zu lesen oder zu kopieren.

ASCII „American Standard Code for Information Interchange“, ein 7-Bit-Zeichensatz, der heutzutage von praktisch allen Computern verstanden wird. Leider enthält ASCII zwar Steuersequenzen wie z. B. Zeilenende und Wagenrücklauf (deren Interpretation von Betriebssystem zu Betriebssystem unterschiedlich ist) und „Klingelton“, aber keine Umlaute und ähnliche Sonderzeichen. Das führt immer noch dazu, daß der Austausch von Texten mit Sonderzeichen zwischen verschiedenen Rechnersystemen mit Problemen behaftet ist.

asymmetrische Verschlüsselung Vgl. „Verschlüsselung“.

Chiffrat Verschlüsselter Klartext.

CRC CRC steht für „cyclic redundancy check“. CRC-Summen sind Prüfsummen, die schnell zu berechnen sind und sehr zuverlässig auf zufällige Änderungen in den Eingabedaten reagieren. Sie werden beispielsweise benutzt, um bei der Datenübertragung mit Modems Übertragunsgfehler zu entdecken. Für kryptographische Zwecke sind CRC-Prüfsummen leider nicht zu gebrauchen, da es sehr einfach ist, zu einer gegebenen Prüfsumme Nachrichten zu konstruieren.

Daten Alles, was ein Computer verarbeiten kann, sind Daten. Daten sind (meist formalisierte) Beschreibungen einer (realen oder eingebildeten) Realität. Beispiele sind dieser Text, eine Bilddatei oder die Angaben einer Statistik.

digitale Unterschrift Eine Abart der asymmetrischen Verschlüsselung, bei der das Chiffrat mit einem geheimen Schlüssel erzeugt wird, so daß mit Hilfe des öffentlichen Schlüssels überprüft werden kann, daß die Nachricht tatsächlich von der autorisierten Person verschlüsselt und somit unterschrieben wurde. Wird i. A. in Verbindung mit digitalen Fingerabdrücken verwendet.

Fingerabdruck Ein mit Hilfe eines festgelegten Verfahrens aus einem beliebig langen Klartext berechneter Wert einer festen Länge. Bei den in der Kryptographie üblichen Verfahren (wie in PGP eingesetzt) läßt sich aus dem Fingerabdruck der Klartext nicht bestimmen und es ist auch nicht möglich, zwei Klartexte mit demselben Fingerabdruck zu berechnen. Näheres finden Sie in Abschnitt 4.6 auf Seite 30.

Information „Information is a difference that makes a difference.“ (GREGORY BATESON) – „Information ist ein Unterschied, der eine Bedeutung hat.“ oder auch „Information ist ein Unterschied, der einen Unterschied macht.“ Nach dieser Definition werden Daten dadurch zur Information, daß sie beim Empfänger eine Entscheidung beeinflussen. Das ist konsistent mit der Definition nach SHANNON, wonach der Informationsgehalt eines Datums (Datum ist die Einzahl von Daten) sich nach der Anzahl der Ja/Nein-Fragen bestimmen läßt, die der Empfänger erst nach Kenntnis dieses Datums beantworten kann.

Eine andere Definition bezieht sich auf den Bedeutungsinhalt der betrachteten Daten. Hiernach ist Information streng an ein Bewußtsein gekoppelt; Computer können lediglich Daten verarbeiten und ihre menschlichen Benutzer bei der Betrachtung der Daten unterstützen, um Informationen zu gewinnen. Sie können aber selbst keine Informationen verarbeiten.

Auf jeden Fall ist Information immer etwas empfängerabhängiges. Den absoluten Informationsgehalt einer Nachricht bestimmen zu wollen, ist etwa genauso sinnvoll wie allgemein angeben zu wollen, wie lange es dauert, ein bestimmtes Buch zu lesen.

geheimer Schlüssel Bei einem asymmetrischen Verfahren derjenige Teil des Schlüsselpaares, der für die Entschlüsselung bzw. die Signatur einer Nachricht notwendig ist.

Klartext Mit Klartext ist im Zusammenhang mit Verschlüsselung von Computerdaten nicht nur Text im Sinne von „für den Menschen lesbar“ gemeint, sondern alle nicht verschlüsselten Daten. PGP verschlüsselt im Allgemeinen einzelne Dateien, so daß „Klartext“ in diesem Handbuch und allgemein bei der Benutzung PGPs als „unverschlüsselte Datei“ gelesen werden kann.

Kryptanalyse Methoden und Verfahren, um chiffrierte Daten ohne vorherige Kenntnis des Schlüssels zu entschlüsseln.

Kryptologie Die Lehre von der Verschlüsselung.

Kryptographie Die praktische Anwendung der Kryptologie.

Mantra In diesem Buch: Der geheime Text, mit dem Ihr privater Schlüssel geschützt ist und den Sie für die Benutzung Ihres Schlüssels eingeben müssen – gewissermaßen die Langversion eines Paßwortes. Näheres zum Mantra und zur Wahl eines guten Mantras steht auf den Seiten 34, 34, 65 und 169.

Nachricht Mit „Nachricht“ ist im Rahmen dieses Handbuches eine einzelne – verschlüsselte oder unverschlüsselte – Datei gemeint. (Teilweise sehen Sie keine Datei im üblichen Sinne, aber auch eine verschlüsselte E-Mail stellt im Wesentlichen eine Datei dar.)

öffentlicher Schlüssel Bei einem asymmetrischen Verfahren das Gegenstück zum geheimen Schlüssel, also der Teil des Schlüsselpaares, der für die Verschlüsselung einer Nachricht bzw. die Überprüfung einer Unterschrift notwendig ist.

privater Schlüssel Siehe „geheimer Schlüssel“.

Prüfsumme Siehe „Fingerabdruck“.

RfC „Request for Comment“, wörtlich „Bitte um Kommentare“. Dieser Name ist leicht irreführend: Die RfCs sind die Sammlung der Definitionen und technischen Spezifikationen, nach denen u. a. UseNet und Internet ablaufen. Sie finden die RfCs gesammelt unter `www.ietf.org/rfc.html`

Schlüssel Ein Verschlüsselungsverfahren hat immer zwei Eingaben: Zum Einen den Klartext (bzw. im Fall der Entschlüsselung das

Chiffrat), zum anderen den Schlüssel. Der Schlüssel dient als Geheimnis, ohne das aus dem Chiffrat der Klartext nicht gewonnen werden kann.

Verschlüsselung Verschlüsselung hat das Ziel, einen Klartext unter Verwendung eines Schlüssels in ein Chiffrat umzuwandeln, so daß es bei Kenntnis des passenden Schlüssels leicht ist, aus dem Chiffrat wieder den Klartext zu gewinnen, es aber nicht möglich (im Sinne von durchführbar) ist, dies ohne den Schlüssel zu tun. Bei den meisten Verfahren ist es darüber hinaus wichtig, daß es auch nicht möglich ist, aus der Kenntnis von Klartext und Chiffrat den Schlüssel zu berechnen. Die für die Ent- und Verschlüsselung verwendeten Schlüssel können identisch sein, dann spricht man von einer *symmetrischen Verschlüsselung,* oder sie können verschieden sein, in diesem Fall liegt eine *asymmetrische Verschlüsselung* vor. Näheres finden Sie in Abschnitt 2.5 ab Seite 9.

H. Index