Nichttechnische Einführung und FAQ's zu PGP

[3]Non-Technical PGP (Pretty Good Privacy) FAQ
by
Andre Bacard, Author of
Computer Privacy Handbook ("The Scariest Computer Book of the Year")
[FAQ Updated November 5, 1996]

Translated by [4]Lutz Donnerhacke to spread this knowledge. This translation is GPLed.
Last changes: 17.12.1997

Die geplante [5]Kryptoregulierung der Bundesregierung kann die Benutzung dieser Software für ILLEGAL erklären!

Dieser Text stellt eine nichttechnische Einführung zu PGP dar, die Dir bei der Entscheidung helfen soll, diese weltweit beliebte Computersoftware zum Schutz Deiner Dateien und eMail einzusetzen oder nicht. Ich habe ihn speziell für humorvolle Personen geschrieben. Diese (unveränderte) FAQ kann für den nichtkommerziellen Gebrauch frei verbreitet werden.

Was ist PGP?

PGP (ausgeschrieben "Pretty Good Privacy" (ziemlich gute Privatsphäre)) ist ein Computerprogramm, das Daten verschlüsselt (unlesbar macht) und entschlüsselt (lesbar macht). Beispielsweise kann PGP "Andre" verschlüsseln, was "457mRT%$354." ergibt. Dein Rechner entschlüsselt den Datenmüll zurück zu "Andre", wenn Du PGP hast.

Wer hat PGP geschrieben?

Philip Zimmermann schrieb das erste Programm. Phil -ein Held für viele Unterstützer des Datenschutzes- arbeitet als Sicherheitsexperte in Boulder, Colorado. Andere Programmierer aus der ganzen Welt haben an den nachfolgenden PGP Versionen und Benutzungsoberflächen mitgeholfen.

PGP benutzt das RSA Public-Key Verschlüsselungssystem. RSA wurde 1977 von seinen Erfindern vorgestellt: Ronald Rivest vom MIT, Adi Shamir vom Weizmann Institute in Israel und Leonard Adelman vom USC. Nach den Anfangsbuchstaben der Nachnamen dieser Leute wird es "RSA" genannt. PGP verwendet außerdem ein Verschlüsselungssystem namens IDEA welches 1990 von Xuejia Lai und James Massey entwickelt wurde.

Wer benutzt PGP Verschlüsselung (oder andere RSA-basierte Systeme)?

Leute, die Wert auf ihre Privatsphäre legen; Politiker, die Wahlkampagnen durchstehen; Steuerzahler, die ihre Steuererklärung abspeichern; Ärtze, die Patientendaten schützen; Unternehmer, die Firmengeheimnisse wahren müssen; Journalisten, die ihre Informanten schützen, und Menschen, die Kontakte suchen, sind nur einige wenige der gesetzestreuen Bürger, die PGP benutzen, um ihre Dateien und eMail geheim zu halten.

Geschäftleute benutzen ebenfalls PGP. Stell Dir vor, Du bist ein Geschäftsführer und mußt per eMail einen Angestellten nach seinem Vorankommen befragen. Du könntest gesetzlich verpflichtet sein, diese eMails geheimzuhalten. Stell Dir vor, Du bist eine Verkäuferin und Du

mußt mit Deiner Zentrale über ein öffentliches Computernetzwerk hinweg, Deine Kundenliste pflegen. Deine Firma und der Gesetzgeber kann fordern, daß diese Liste geheim bleibt. Dies sind nur einige wenige Gründe weswegen im Geschäftsleben Verschlüsselung zum Schutz der Kunden, der Angestellten und der Firmen selbst eingesetzt wird.

PGP hilft auch bei sichere Finanzübertragungen. Bspw. die Electronic Frontier Foundations (EFF) benutzt PGP zur Verschlüsselung von Mitgliedernummern, so daß die Mitglieder per eMail ihre Beiträge bezahlen können.

Thomas G. Donlan, ein Redakteur bei _Barron's_ (einer Finanzpublikation, die zum _The Wall Street Journal_ gehört), schrieb ein ganzseitiges Editorial am 25.April 1994. _Barron's_ übertitelte es mit "Privacy and Security: Computer Technology Opens Secrets, And Closes Them." (Datenschutz und Datensicherheit: Computertechnik öffnet Geheimnisse und verschließt sie)

Mr. Donlan schrieb (auszugsweise):

> "RSA Data Security, the company founded by the three inventors, has hundreds of satisfied customers, including Microsoft, Apple, Novell, Sun, AT&T and Lotus. Versions of RSA are available for almost any personal computer or workstation, many of them built into the operating systems. Lotus Notes, the network communications system, automatically encrypts all it messages using RSA. Other companies have similar products designed around the same basic concept, and some versions are available for free on computer bulletin boards."

> RSA Data Security, gegründet von den drei Erfindern, hat Hunderte von zufriedenen Kunden, zu denen auch Microsoft, Apple, Novell, Sun, AT&T und Lotus gehören. Implementationen von RSA stehen in fast jedem Personalcomputer oder Workstation zur Verfügung, oft ins Betriebssystem eingebaut. Lotus Notes, das Netzkommunikationssystem, verschlüsselt automatisch alle Nachrichten mit RSA. Andere Firmen haben vergleichbare Produkte, entwickelt nach dem gleichen Prinzip, und einige Versionen sind in Mailboxen frei verfügbar.

Donlan fährt fort:

> "Without security, the Internet is little more than the world's biggest bulletin board. With security, it could become the information supermarket of the world. RSA lets people and banks feels secure putting their credit-card numbers on the public network. Although it still seems that computers created an age of snoopery, the age of privacy is at hand."

> Ohne Sicherheit ist das Internet nicht viel mehr als die weltgrößte Mailbox. Mit Sicherheit kann es der Informationssupermarkt der Zukunft werden. RSA gestattet es Menschen und Banken, sich bei der Angabe der Kreditkartennummer im öffentlichen Netz sicher zu fühlen. Obwohl es noch den Anschein hat, daß Computer ein Zeitalter der Schnüffelei begannen, liegt das Zeitalter der Privatsphären vor uns.

Sind nicht Computer und eMail schon sicher?

Deine Computerdateien (ohne Verschlüsselung) können von jedem gelesen werden, der Zugriff auf Deinen Rechner hat. EMail ist notorisch

unsicher. Die typische eMail wandert über viele Systeme. Die Personen, die diese Systeme betreiben, können Deine Nachrichten lesen, kopieren und speichern. Viele Mitbewerber und Voyeure sind hochmotiviert eMail abzufangen. Eine geschäftliche, offizielle oder persönliche Nachricht mit dem Computer zu versenden, ist weniger geschützt als das gleiche auf einer Postkarte zu verschicken. PGP ist ein sicherer "Briefumschlag", der Mitbewerber und Kriminelle abhält, Dir zu schaden.

Ich habe nichts zu verbergen. Wozu brauche ich Datenschutz?

Zeig mir einen Menschen, der keine Geheimnisse vor seiner Familie, seinen Nachbarn oder seinen Kollegen hat und ich zeige Dir jemanden, der entwerder ein ungewöhnlicher Exhibitionist oder ein unglaublicher Dummkopf ist.

Zeige mir eine Firma, die keine Firmengeheimnisse hat und ich zeige Dir ein Geschäft, das nicht läuft.

In einem Brief schrieb mir ein Student folgendes:

> "I had a part-time job at a dry cleaner. One day I returned a diamond ring that I'd found in a man's coat pocket to his wife. Unfortunately, it was NOT her ring! It belonged to her husband's girlfriend. His wife was furious and divorced her husband over this incident. My boss told me: 'Return jewelry ONLY to the person whose clothes you found it in, and NEVER return underwear that you find in pockets!' Until that moment, I thought my boss was a finicky woman. But she taught me the need for PGP."

> Ich hatte einem einen Teilzeitjob als Reinigungskraft. Eines Tages brachte ich einen Diamantring, den ich in einer Manteltasche eines Mannes gefunden hatte, seiner Frau zurück. Unglücklicherweise war es NICHT ihr Ring! Er gehörte der Freundin ihres Mannes. Seine Frau wurde fuchsteufelswild und ließ sich wegen dieses Vorkommnisses scheiden. Mein Chef sagte mir: 'Gib Juwelen NUR der Person zurück, in deren Sachen Du es fandest. Und gib NIEMALS Unterwäsche zurück, die Du in einer Tasche fandest.' Bis zu diesem Moment dachte ich, mein Chef wäre eine engstirnige Frau. Aber sie lehrte mich die Notwendigkeit von PGP.

Geheimhaltung, Diskretion, Vertraulichkeit und Besonnenheit sind die Markenzeichen dieser Zivilisation.

Ich habe gehört, die Polizei sage, daß Verschlüsselung verboten werden soll, weil Kriminelle sie zur Tarnung benutzen. Ist das wahr?

Das nächste Mal, wenn Du jemanden das erzählen hörst, frag ihn, ob er Thomas Jefferson verbieten möchte, den "Vater der amerikanischen Kryptographie", der die amerikanische _Unabhängigkeitserklärung_ schrieb.

Viele Regierungen, Firmen und Gesetzeshüter benutzen Verschlüsselung um ihre Operationen zu verstecken. Ja, einige Kirminelle benutzen auch Verschlüsselung. Kriminelle benutzen aber häufiger Autos, Handschuhe und Masken, um zu entkommen.

PGP ist "Verschlüsselung für die Massen". Es gestattet dem gesetzestreuen Durchschnittbürger einige der Schutzmaßnahmen, die die Regierungen und Firmen für sich beanspruchen.

Wie arbeitet PGP?

PGP ist ein "öffentliches Schlüsselsystem" (public key cryptography). Wenn Du PGP das erste Mal startest, generiert es zwei "Schlüssel", die nur zu Dir gehören. Stell Dir diese Schlüssel wie ein Gegenstück zu den Schlüssel Deiner Tasche vor. Ein PGP-Schlüssel ist der GEHEIME Schlüssel und bleibt auf Deinem Rechner. Der andere Schlüssel ist ÖFFENTLICH. Du gibst diesen öffentlichen Schlüssel Deinen Gesprächspartnern. Hier ist ein Beispiel eines ÖFFENTLICHEN Schlüssels:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.7

mQA9Ai2wD2YAAAEBgJ18cV7rMAFv7P3eBd/cZayI8EEO6XGYkhEO9SLJOw+DFyHg
Px5o+IiR2A6Fh+HguQAFEbQZZGVtbyA8ZGVtb0B3ZWxsLnNmLmNhLnVzPokARQIF
EC2wD4yR2A6Fh+HguQEB3xcBfRTi3D/2qdU3TosScYMAHfgfUwCelbb6wikSxoF5
ees9DL9QMzPZXCioh42dEUXP0g==
=sw5W
-----END PGP PUBLIC KEY BLOCK-----
```

Stell Dir vor, der ÖFFENTLICHE Schlüssel gehört zu Dir und Du mailst ihn mir. Ich kann Deinen ÖFFENTLICHEN Schlüssel in meiner PGP Software speichern und benutze Deinen ÖFFENTLICHEN Schlüssel, eine Nachricht zu verschlüsseln, die nur Du lesen kannst. Einer der Vorteile von PGP ist, daß Du mir diesen Schlüssel geben kannst, wie Deine Telefonnummer. Wenn ich Deine Telefonnummer habe, kann ich Dich anrufen; aber ich kann Dein Telefon nicht abheben. Ähnlich ist es mit dem ÖFFENTLICHEN Schlüssel: Ich kann Dir eine Nachricht verschlüsseln; Ich kann sie aber nicht lesen.

Dieses Konzept des ÖFFENTLICHEN Schlüssels kann anfangs etwas seltsam klingen. Trotzdem wird es völlig klar, sobald Du etwas mit PGP rumspielst.

Wie sicher ist PGP? Schützt es mich wirklich?

Vielleicht kann die Regierung oder Deine Schwiegermutter PGP Nachrichten "knacken", indem sie Supercomputer oder Genialität einsetzt. Davon habe ich keine Ahnung. Drei Dinge sind jedoch sicher:

1. Hervorragende Kryptoanalytiker und Computerexperten haben vergeblich versucht, PGP zu knacken.
2. Wer auch immer nachweist, daß er PGP enträtselt hat, wird schnell zu Ruhm unter den Kryptographen kommen. Er wird viel Beifall ernten und eine Menge Geld angeboten bekommen.
3. Die PGP Programmierer werden es sofort bekanntgeben.

Fast täglich verbreitet jemand eine Nachricht, wie "PGP durch Jugendlichen aus Omaha geknackt". Bezweifle diese Aussagen. Die Welt der Kryptographie zieht Paranoiker, Provokateure und Aliens in ihren Bann.

Bis heute hat niemand öffentlich vorgeführt, PGP überlistet oder geknackt zu haben.

Ist PGP legal in den USA?

Ja.

+++ Wichtige Bemerkung +++. Es ist verboten, PGP aus den USA zu exportieren. Tue dies niemals! Um mit Freunden außerhalb der USA (bspw. in England) zu kommunizieren, bitte sie, sich PGP von Quellen außerhalb der USA zu besorgen.

Ist PGP legal außerhalb der USA?

Die Verwendung von PGP wird in den meisten Staaten erlaubt. Zusätzlich ändern sich die Gesetze ständig auf der ganzen Welt. Prüfe die lokalen Gesetze in Deinem Heimatland. [In Europa ist PGP frei einsetzbar, abgesehen von Frankreich und Russland -LD]

Was ist eine digitale Unterschrift?

Stell Dir vor, ich unterschreibe diese FAQ mit meiner digitalen PGP-"Unterschrift". Das gestattet es Personen, die PGP und meinen ÖFFENTLICHEN Schlüssel haben zu überprüfen, daß

1. ich -Andre Bacard- (und nicht der Star aus der Sportzeitung, der behauptet ich zu sein!) dieses Dokument geschrieben habe.
2. niemand das Dokument verändert hat, seit ich es unterschrieben habe.

PGP Unterschriften sind hilfreich bei Vertragsunterzeichnungen, Geldanweisungen und bei der Personenidentifizierung.

Wie schwierig ist es, PGP zu erlernen?

PGP ist deutlich leichter zu bedienen, als bspw. eine Textverarbeitung.

Ist PGP für meinen Computer verfügbar?

Es gibt Versionen für DOS und Windows, ebenso für verschiedenen Unixe, Macintosh, Amiga, Atari ST, OS/2 und CompuServes WinCIM und CSNav. Viele Leute arbeiten daran, diese Verfügbarkeit zu erweitern. Lies die Usenet Newsgruppe alt.security.pgp, um den neuesten Stand zu erfahren.

Sind diese Versionen von PGP untereinander kompatibel?

Ja. Bspw. ist ein mit PGP unter DOS verschlüsseltes Dokument von jemanden lesbar, der PGP unter Unix einsetzt.

Ab 1.September 1994 kann die Version 2.6 und höher frühere Versionen lesen, jedoch können Versionen vor der 2.6 nicht länger die neuen Versionen lesen. Ich lege jedem ans Herz, auf die Versionen 2.6.2 oder 2.6.3 upzudaten.

Seit Mitte 1997 gibt es die Versionen 5.x. Diese sind zu den 2.6.x Versionen i.d.R. inkompatibel. Bis zur Klärung der Kontroversen um die in diesen Versionen eingebauten Mitlesemöglichkeiten durch Dritte (i.d.R. Firmen) empfehle ich, bei 2.6.3 zu bleiben.

Woher bekomme ich PGP?

Wenn Du in Amerika bist, gehe zum [6]MIT oder zur [7]PGP, Inc.. Wenn Du außerhalb der USA bist, sehe auf der [8]Internationalen PGP-Seite nach.

PGP gibt es auch auf vielen Mailboxen (BBS) und von FTP("File Transfer Protocol")-Server auf aller Welt. Diese Server -wie Videotheken- kommen und gehen.

Michael Johnson pflegt eine hervorragende FAQ, die die aktuellen Server auflistet, auf denen PGP zu bekommen ist. Er postet diese FAQ regelmäßig nach alt.security.pgp.

Wie teuer ist PGP?

Die PGP Versionen, die Du in Mailboxen oder auf FTP-Servern findest sind "Freeware". Das bedeutet, sie sind frei. Personen von Neuseeland bis Mexiko benutzen diese Versionen täglich. Je nachdem wo Du wohnst, verletzt diese "Freeware" lokale Gesetze oder nicht.

Noch etwas, was ich wissen sollte?

DEINE Privatsphäre und Sicherheit können gefährdert sein! Umfangreiche Banken-, Kreditkarten- und medizinische Datenbanken, eMail Überwachung und Computerschnüffelprogramme sind nur einige wenige Faktoren, die jeden gesetzestreuen Bürger treffen. Kurz gesagt, unsere in Privatsachen schnüffelnde Gesellschaft dient den Kriminellen und serviert Deine Daten auf einem Silbertablett.

[1]zurück

Die Comp.security.pgp FAQ

Übersetzung der Version 1.5

Dies ist die Liste der häufig gestellten Fragen für das Verschlüsselungsprogramm "Pretty Good Privacy" (PGP) von Phillip Zimmermann und anderen. Der englische Originaltext von [2]galactus@stack.nl wird einmal im Monat in allen comp.security.pgp-Newsgruppen veröffentlicht und ist auch im World wide web (www) unter [3]http://www.pgp.net/pgpnet/pgp-faq/ verfügbar.

Siehe "[4]über diese FAQ" für mehr Informationen. Der Abschnitt "[5]Was ist neu" beschreibt Ergänzungen, Änderungen und Löschungen in dieser Version der FAQ.

Inhaltsverzeichnis

Einleitende Fragen

1.1 Was ist PGP?

PGP ist ein Programm, das Deiner elektronischen Post eine Eigenschaft verleiht, die sie sonst nicht hätte: Vertraulichkeit. Dies wird durch Verschlüsselung Deiner E-Mail erreicht, so daß sie von keiner anderen als der berechtigten Person gelesen werden kann. Im verschlüsselten Zustand sieht die Nachricht aus wie ein bedeutungsloses Durcheinander zufälliger Buchstaben. PGP hat seine Fähigkeit bewiesen, sogar den ausgeklügeltesten Analyseversuchen zu widerstehen. PGP kann auch verwendet werden, um eine digitale Signatur anzubringen, ohne die Nachricht zu verschlüsseln. Diese Funktion wird normalerweise in öffentlichen Newsbeiträgen verwendet, wo Du nicht versteckst, was Du zu sagen hast, sondern vielmehr Anderen erlauben willst, Deine Urheberschaft zu überprüfen. Wenn die digitale Signatur erst einmal erzeugt ist, kann niemand mehr die Nachricht oder die Signatur verändern, ohne daß PGP die Manipulation entdeckt.

Während PGP leicht zu benutzen ist, hat es auch seine Tücken: Du solltest Dich mit den verschiedenen Optionen in PGP gründlich vertraut machen, bevor Du es für den Versand wirklich ernster Nachrichten verwendest. Zum Beispiel würde das Kommando "pgp -sat <dateiname>" eine Nachricht nur unterschreiben, nicht verschlüsseln. Obwohl das Ergebnis aussieht, als sei es verschlüsselt, ist es das in Wirklichkeit nicht. Jeder auf der ganzen Welt könnte den originären Text zurückgewinnen.

1.2 Warum sollte ich meine E-mail verschlüsseln? Ich tue nichts Illegales!

Hindert Dich etwas daran, Deine gesamte Korrespondenz auf der Rückseite einer Postkarte zu erledigen? Aus den gleichen Gründen solltest Du Deine Email verschlüsseln. E-Mail ist tatsächlich weit weniger sicher, als das Postsystem. Im Falle der Briefpost steckst Du zumindest Deinen Brief noch in einen Umschlag, um ihn vor beiläufiger Schnüffelei zu bewahren. Schau Dir einmal den Header-Bereich irgendeiner empfangenen E-mail an und Du wirst sehen, daß sie auf dem Weg zu Dir eine Anzahl von Knotenpunkten durchlaufen hat. Jeder Einzelne dieser Knotenpunkte bietet die Möglichkeit zur Spionage. Verschlüsselung sollte niemals auf illegale Aktivitäten hinauslaufen. Sie ist lediglich dazu gedacht, private Meinungen privat zu vermitteln.

Jemand hat es einmal so ähnlich ausgedrückt:

Verbrechen? Wenn Du kein Politiker bist, kein Grundlagenforscher, Finanzier, Vorstandsvorsitzender, Rechtsanwalt, keine Berühmtheit, kein Freidenker in einer unterdrückenden Gesellschaft oder ein Mensch, der zu viel Spaß hat, und wenn Du keine Email verschickst über Dein privates Sexleben, Deine finanziellen / politischen / rechtlichen / wissenschaftlichen Pläne oder Klatsch, dann vielleicht brauchst Du PGP nicht. Aber erkenne wenigstens, daß Privatsphäre nichts mit Verbrechen zu tun hat, daß sie es ist, die tatsächlich unsere Welt davon abhält, auseinanderzufallen. Davon abgesehen, macht PGP SPAß. Du hattest niemals einen geheimen Decoder-Ring (kleines Kinderspielzeug in amerikanischen Kornflakes-Packungen)? Boo!

1.3 Was sind öffentliche und private Schlüssel?

Bei konventionellen Verschlüsselungsmethoden, müssen Schlüssel auf "anderem" Wege ausgetauscht werden, bei Treffen unter vier Augen oder über vertrauenswürdige Kuriere. Das Problem ist, daß ein sicherer Informationsaustausch schon erfolgen muß, bevor Du einen sicheren Austausch vornehemen kannst! Bei der konventionellen Kryptographie wird entweder der selbe Schlüssel für Ver- und Entschlüsselung benutzt, oder es ist auf einfachem Wege möglich, den einen in den anderen Schlüssel zu konvertieren. Bei der Verschlüsselung mittels "öffentlicher Schlüssel" unterscheiden sich die "Chiffrier-" und "Dechiffrier-Schlüssel", und es ist niemandem möglich, den einen in den anderen umzuwandeln. Darum kann der Chiffrier-Schlüssel öffentlich zugänglich gemacht werden und irgendwo in eine Datenbank geschrieben werden. Jeder, der Dir eine Nachricht senden möchte, würde Deinen Chiffrierschlüssel aus dieser Datenbank oder anderswoher beziehen und seine Nachricht an Dich verschlüsseln. Die Nachricht kann nicht mit dem Chiffrierschlüssel entziffert werden! Daher kann niemand, außer dem dazu berechtigten Empfänger die Nachricht entschlüsseln. Noch nicht einmal die Person, die den Text verschlüsselt hat, kann den Prozeß umkehren. Wenn Du eine Nachricht empfängst, benutzt Du Deinen privaten Dechiffrier-Schlüssel um die Nachricht zu entziffern. Dieser geheime Schlüssel verläßt niemals Deinen Computer. Tatsächlich ist Dein geheimer Schlüssel selbst verschlüsselt um ihn vor jedem, der an Deinem Computer herumschnüffelt, zu schützen.

1.4 Wieviel kostet PGP?

Nichts!

Es sollte dennoch beachtet werden, daß einige Freeware-Versionen von PGP in den USA gegen ein Patent der "Public Key Partners" (PKP) verstoßen. Speziell die MIT- und ViaCrypt-Versionen sind _keine_ ungesetzlichen Programme; solltest Du innerhalb der USA eine andere Version benutzen, tust Du das auf eigenes Risiko. Weitere Informationen bezüglich Patenten siehe unten ([127]1.6). Davon abgesehen sind Freeware-Versionen von PGP nur fur den nicht-kommerziellen Gebrauch kostenlos. Solltest Du PGP in einem kommerziellen Umfeld anwenden (und in den USA oder Kanada wohnen), solltest Du Dir ViaCrypt PGP besorgen. ViaCrypt PGP hat noch andere Vorteile, zu vorderst die beschränkte Lizenz zur Verbreitung in Zweigstellen. Wie Du mit ViaCrypt in Kontakt trittst, steht unten unter Frage [128]1.9. Solltest Du PGP für kommerzielle Zwecke außerhalb der USA verwenden, solltest Du Dich an die Ascom Systec AG wenden; sie hält das IDEA-Patent und verkauft Lizenzen für die Benutzung der (konventionellen) IDEA-Verschlüsselung in PGP. Kontaktadresse:

Erhard Widmer

Ascom Systec AG
Dep't. CMVV
Gewerbepark
CH-5506 Mägenwil
Switzerland
[129]IDEA@ascom.ch
Tel: ++41 64 56 59 83
FAX: ++4164 56 59 90

1.5 Ist Verschlüsselung legal?

In einem Großteil der zivilisierten Welt ist Verschlüsselung entweder legal, oder sie wird zumindest toleriert. Trotzdem gibt es einige Länder, wo derartige Praktiken Dich vor ein Erschießungskommando bringen können! Mache Dich mit der Gesetzgebung in Deinem eigenen Land vertraut, bevor Du PGP oder irgend eine andere Verschlüsselungssoftware benutzt. Einige der Länder, in denen Verschlüsselung illegal ist, sind Frankreich, der Iran, Rußland und der Irak. Die Rechtslage in vielen Ländern wird im WWW auf [130]http://cwis.kub.nl/~frw/piople/koops/lawsurvy.htm dargelegt.

1.6 Ist PGP legal?

Personen, die in den Vereinigten Staaten oder Kanada leben, sollten ergänzend zu den vorher schon gemachten Bemerkungen zur Kryptographie, ein paar zusätzliche Punkte beachten.

Zunächst ist die Frage, ob PGP unter die ITAR-Regelungen fällt oder nicht. ITAR regelt den Export kryptographischer Technologie aus den Vereinigten Staaten. Dies trotz der Tatsache, daß technische Abhandlungen über Verschlüsselungsverfahren seit Jahren schon weltweit verfügbar waren. Jeder kompetente Programmierer wäre fähig gewesen, die Erkenntisse in ein funktionierendes Verschlüsselungsprogramm einfließen zu lassen. Eine Klage der EFF (Electronic Frontier Foundation) gegen die ITAR-Restriktionen könnte zu deren Abschwächung führen und der Erlaubnis, Verschlüsselungstechnologie zu exportieren.

Die Situation in Kanada ist etwas kompliziert; obwohl ITAR hier nicht gilt, respektiert Kanada die US-Exportbeschränkungen. Das bedeutet, der Export von PGP aus Kanada ist ungesetzlich, wenn es vorher aus den USA importiert worden ist.

Außerdem bestand die Ansicht, daß ältere Versionen von PGP (bis Version 2.3a) in den USA die Patentrechte der Public Key Partners (PKP) verletzten. Dies wurde jedoch niemals vor Gericht geklärt und die derzeit gängigen Versionen von PGP sind auf der Basis verschiedener Uebereinkünfte und gültiger Lizenzen entstanden, um den Patentstreit beizulegen. Sogenannte "internationale" und ältere Versionen (vor ViaCrypt 2.4) werden von PKP immernoch als unrechtmässig betrachtet. Wenn Du Dich in den USA befindest, benutzt Du sie auf eigenes Risiko!

1.7 Welche ist die aktuelle Version von PGP?

Im Moment gibt es fünf verschiedene "aktuelle" Versionen von PGP. Alle gehen mehr oder weniger auf eine gemeinsame Urform zurück: PGP 2.3a, der letzten "guerillaware" -Version von PGP. Absprachen, die zum Ziel hatten, PGP zu legalisieren, führten zu den unterschiedlichen Ausführungen. Alle haben im Großen und Ganzen den gleichen Funktionsumfang, und alle arbeiten in der Regel zusammen.

Alle Versionen nach 2.3 erzeugen Formate, die nicht mehr von 2.3 oder älteren Versionen verarbeitet werden können. In den internationalen

Versionen gibt es jedoch den Schalter 'legal_kludge=on', der auch das alte Format erzeugen kann.

MIT PGP 5.0
: Es gibt zwei veröffentlichte Freeware Version von PGP 5.0: Für Windows'95 und für den Macintosh. Diese Version hat einie Beschränkungen, die über die der bisherigen "offiziellen" Freeware Version 2.6.2 hinausgehen. Beispielsweise:

  + Keine konventionelle Verschlüsselung.
  + Kein "sicheres" Überschreiben der Festplatte.

  Die Quellen der 5.0 Version gibt es nur in Buchform. Mit internationalen Bemühungen wird momentan dieser Code eingescannt und korrekturgelesen. Die USA Exportgesetze verbieten zwar die Ausfur der elektronischen Form, nicht aber die Ausfuhr der Buchform.

PGP, Inc
: verkauft zwei Versionen von PGP:

  + PGPmail 4.5 für Geschäftsanwendungen (früher Viacrypt PGP Business Edition) und
  + PGP 5.0 für Privatgebrauch.

  Siehe dazu auch Frage [131]1.9.

PGP 2.6.3i("international")
: ist eine PGP-Version, die aus dem illegal exportierten Quellcode von MIT PGP entwickelt worden ist. Im Grundsatz handelt sich um MIT PGP 2.6.2, aber es benutzt die älteren Verschlüsselungsroutinen aus PGP 2.3a; sie arbeiten besser als RSAREF und außerdem unterliegen sie nicht den Beschränkungen der RSAREF-Lizenz. Es enthält außerdem einige Fehlerkorrekturen, die seit dem Erscheinen von MIT PGP 2.6.2 aufgetaucht sind und kleinere Verbesserungen. Für mehr Informationen kannst Du die "internationale PGP-Homepage" heranziehen: [132]http://www.ifi.uio.no/pgp/

PGP 2.6ui ("unofficial international")
: ist PGP 2.3a mit einigen kleineren Veränderungen, die es erlauben, Dateien zu entschlüsseln, die mit MIT PGP verschlüsselt worden sind. Es enthält keine der MIT-Neuerungen und -Verbesserungen. Allerdings besitzt es andere neue Eigenschaften, besonders in der Macintosh - Version.

PGP 2.6.3(i)n
: ist die Version der [133]Zertifizierungsinstanz des Individual Network e.V.. Sie unterstützt die in der pgformat.doc beschriebenen Fähigkeiten:

  + Verfall von Schlüsseln
  + Rückruf von Schlüsselsignaturen und Nutzerkennungen
  + Angabe der Qualität der Identitäsprüfung bei Schlüsselsignaturen

  Es wurden nervige Fehler von PGP behoben:

  + Schlüsselunterschriften zurückgezogender Schlüssel sind jetzt ungültig

  Neue Fähigkeiten wurden hinzugefügt:

+ Es wird zwischen Unterschriften- und
  Verschlüsselungsschlüssel unterschieden.
+ Unterstützung einer seperaten "Encrypt To Self Id"

1.8 Gibt es ein Archiv der comp.security.pgp -Gruppen?

Eigentlich nicht. Natürlich kannst Du es mit Dejanews ([134]http://www.dejanews.com/) oder Alta Vista ([135]http:/altavista.digital.com/) versuchen, wenn Du Artikel über ein spezielles Thema suchst.

1.9 Gibt es eine kommerzielle Version von PGP?

Ja. Bis vor kurzem vermarktete ViaCrypt die einzige kommerzielle, lizensierte Version von PGP. Die Firma wurde von PGP Inc., einer Gründung Phil Zimmermanns, aufgekauft. Diese Firma verkauft zwei Versionen: PGPmail 4.5 für Geschäftsanwendungen und PGP 5.0 für Privatgebrauch. Es ist unklar, ob die Lizenzen von RSA für ViaCrypt immer noch gelten.

Die [136]PGP 5.0 FAQ diskutiert das aufühlicher.

PGPmail 4.5 ist der Nachfolger von ViaCrypts Business Edition (BE). Zusätzlich zu den Features der Freeware-Versionen bietet dieses Produkt die Installation eines "Corporate Access Key" (eines firmeninternen Generalschlüssels) zu, so daß die Firma Zugriff auf alle verschlüsselten Nachrichten nehmen kann, wenn mal ein privater Schlüssel verlorengeht. Darüberhinaus gibt es einen _Enclyptor_, der eine Komandoleiste in E-Mail- und Textverarbeitungsprogramme hinzufügt. Weitere Informationen finden sich unter [137]http://www.pgp.com/products/PGPmail-faq.cgi

(_Beachte:_ Die "Corporate Access Key" Funktion von PGPmail 4.5 stellt keine Hintertür zu PGP dar. Die Freeware Versionen haben diese Funktionalität nicht. Auch schwächt diese Funktion nicht die eigentliche Verschlüsselung, sie bietet nur der Firma Zugriff auf die Daten ihrer Mitarbeiter, so diese ihren privaten Schlüssel verloren oder das Mantra vergessen haben.)

1.10 Ist PGP in Form einer Programmbibliothek verfügbar, so daß ich Programme schreiben kann, die darauf zugreifen?

PGP 3.0 wird, wenn es fertig ist, diese Anforderungen voraussichtlich erfüllen. Das PGP-Entwicklungsteam hat sogar eine vorläufige Anwendungsschnittstelle für diese Bibliothek herausgegeben; Du bekommst sie von [138]ftp://ftp.pgp.net/pub/pgp/doc/950212_pgp3spec.txt.gz

Das Entwicklerteam hat klargestellt, daß dies kein endgültiges Exemplar ist; Einiges daran ist bereits veraltet. Dennoch ist es geeignet, sich einen Überblick zu verschaffen. Sende Kommentare an [139]pgp@lsd.com.

Es gibt eine PGP Bibliothek in einem frühen Entwicklungsstadium: [140]ftp://dslab1.cs.uit.no/pub/PGPlib-0.1.tar.gz.

Alternativ kannst Du Deine Programme so schreiben, daß Du PGP aufrufst, wenn es notwendig ist. In C würdest Du bspw. system() oder spawn..() verwenden.

Einige Leute arbeiten an einer DLL-Version (häufig für Windows 3.1 oder NT) von PGP, aber ich habe keinerlei Informationen über den Status dieser Versionen. PGP Inc. (früher ViaCrypt, siehe Frage

[141]1.9) verkauft eine MS Windows DLL, die alternativ verwendet werden kann.

1.11 Mit welchen Betriebssystemen arbeitet PGP?

PGP wurde erfolgreich auf verschiedene Plattformen portiert, darunter DOS, Macintosh, OS/2, Unix (fast alle Geschmacksrichtungen), VMS, Atari ST, Acorn RISC OS (Archimedes) und den Commodore Amiga. Ein Windows NT-Kompilat ist angeblich ebenfalls in Arbeit. Wenn Deine bevorzugte Plattform hier nicht aufgeführt wird, nicht verzweifeln! Es dürfte wahrscheinlich nicht allzu schwierig sein, PGP auf Dein Betriebssystem zu übertragen, wenn man bedenkt, was in dieser Hinsicht schon geleistet worden ist. Frage einfach herum, ob es vielleicht schon eine Version gibt, und wenn nicht, versuche es!

Die VMS-Version von PGP hat übrigens ihre eigene Web-Seite: [142]http://www.tditx.com/~d_north/pgp.html

1.12 Wo bekomme ich PGP?

PGP ist sehr weit verbreitet, so sehr, daß eine eigene FAQ geschrieben worden ist, um diese Frage zu beantworten. Sie heißt:"Where To Get The Pretty Good Privacy Program (PGP)"; sie wird regelmäßig in alt.security.pgp veröffentlicht, steht in den verschiedenen FAQ-Archiven und kann auch von [143]ftp://ftp.csn.net/mpj/getpgp.asc bezogen werden.

Trotzdem werde ich beschreiben, wie man an die verschiedenen Versionen von PGP herankommt. Greife bitte auf das erwähnte Dokument zurück, um an mehr Informationen zu gelangen.

MIT PGP

Wegen der ITAR-Regeln hielt es MIT für notwendig, PGP in einem export-kontrollierten Verzeichnis unterzubringen, um Menschen außerhalb der USA daran zu hindern, die Dateien herunterzuladen.

Wenn Du Dich innerhalb der USA befindest, folge diesen Hinweisen: Telnetverbindung nach net-dist.mit.edu und log-in mit "getpgp". Du wirst eine kurze Erläuterung zur der Regulierung im Export kryptographischer Software erhalten, dann mußt Du eine Reihe von JA/NEIN - Fragen beantworten. Wenn Du korrekt auf alle Fragen antwortest (Es sind größtenteils Einverständniserklärungen bezüglich der RSADSI- und MIT-Lizenzen und Fragen nach möglicher Exportabsicht); man wird Dir einen speziellen Verzeichnisnamen nennen, in dem Du den PGP-Code findest. Jetzt kannst Du die FTP-Verbindung zu net-dist.mit.edu herstellen, in das besagte Verzeichnis wechseln und die Software herunterladen. Der Zugang zu den Verzeichnissen kann Dir verwehrt werden, obwohl Du alle Fragen korrekt beantwortet hast, wenn der MIT-Server nicht bestätigen kann, daß Dein Rechner sich in der Tat in den USA befindet.

Weitere Hinweise, Formulierungen der MIT und RSAREF-Lizenzen, Bemerkungen und die volle Dokumentation sind frei erhälltlich auf [144]ftp://net-dist.mit.edu/pub/PGP/

Ein einfacherer Weg an die PGP-Software zu gelangen, ist die folgende Adresse: [145]http://bs.mit.edu:8001/pgp-form.html

PGPmail und PGP 5.0

Die Freeware Version gibt es beim MIT (siehe oben). Die anderen Versionen sind kommerzielle Software, die von PGP Inc. gekauft

werden müssen. Abgesehen davon sind sie außerhalb der USA oder Kanadas nur unter bestimmten Umständen erhältlich (Kontaktadressen siehe oben, Frage [146]1.9).

PGP 2.6.3i

Weil Norwegen nicht den ITAR-Beschränkungen unterliegt, sind keinerlei Verrenkungen nötig, um an diese Version zu kommen: [147]http://www.ifi.uio.no/pgp/

Du kannst es auch über E-Mail beziehen,indem Du eine E-mail an [148]hypnotech-request@fi.uio.no schickst, mit Deiner Bitte im Subject-Header: GET pgp262i(s).(zip|tar.gz). Wähle den Buchstaben "s", wenn Du den Quellcode haben willst. Mit "zip" am Ende erhälst Du die Dateien im PKZIP/Info-ZIP Archiv-Format, "tar.gz" dagegen bedeutet gzipped tar file. Eine US-kompilierte Fassung von 2.6.3i (was bedeutet, daß der MPILIB RSA-Alrgorithmus nicht benutzt wird, und amerikanische Patente nicht berührt werden) kann von [149]http://www.isc.rit.edu/~pdw5973/downinst.html. bezogen werden.

PGP 2.6ui:

[150]ftp://ftp.mantis.com.uk/pub/cryptography/

[151]http://www.mantis.co.uk/pgp/pgp.html
Dieser Link ist auch eine exzellente Quelle zusätzlicher Informationen über PGP.

PGP 2.6.3(i)n:

[152]ftp://ftp.iks-jena.de/pub/mitarb/lutz/crypt/software/pgp/

Bemerkungen über Ftpmail: Leute ohne Zugang zu FTP, jedoch mit E-Mail-Anschluß können FTP-Dateien zugeschickt bekommen. Um Informationen über diesen Dienst zu erhalten, mußt Du "Help" als Nachricht an [153]ftpmail@ftpmail.ramona.vix.com. senden. Man wird Dir eine Anleitung zuschicken.

1.13 Ich will noch mehr wissen!

Sollte diese FAQ Deine Fragen nicht beantworten, gibt es viele Stellen mit Informationen über PGP.

World Wide Web:

[154]http://sun1.bham.ac.uk/N.M.Queen/pgp.html
ein guter Startpunkt, enthält Links zum Herunterladen von PGP.

[155]http://www.stack.nl/~galactus/remailers/bg2pgp.txt
Obwohl die Dokumentation im Lieferumfang von PGP bereits sehr umfassend ist, willst Du vielleicht diese Anleitung lesen. Sie befaßt sich mit den grundlegenden Schritten der Installation und des Gebrauchs von PGP und enthält Tips zur effektiveren Nutzung.

[156]http://www.stack.nl/~galactus/remailers/passphrase-faq.html
Dein Paßphrase (auch: Mantra) wird benutzt, um Deinen geheimen PGP-Schlüssel zu schützen. Hier steht, wie man sichere Paßphrases generiert und handhabt. Das kann auch bei Paßwörtern für andere Zwecke nützlich sein.

[157]http://www.stack.nl/~galactus/remailers/attack-faq.html

Eine sehr detaillierte Analyse der Sicherheit von PGP und möglicher Angriffsszenarien.

FTP-Ressourcen:

[158]ftp://ftp.pgp.net/pub/pgp/
[159]ftp://ftp.ox.ac.uk/pub/crypto/
[160]ftp://ripem.msu.edu/pub/crypt/
[161]ftp://ftp.csua.berkeley.edu/pub/cypherpunks/

Siehe auch Teil [162]10, "Weiterführende Literatur".

Sehr allgemeine Fragestellungen und Probleme

2.1 Warum kann jemand, die Version 2.3 verwendet, meine Version 2.6-Nachrichten nicht lesen?

Du benutzt wahrscheinlich MIT PGP, oder vielleicht eine andere PGP-Version mit ausgeschalteter "legal_kludge"-Option. Als Teil der Einverständniserklärung, die die Patentstreitigkeiten beilegen sollte, hat MIT PGP das Format etwas abgeändert, um es PGP 2.4 und älteren Versionen unmöglich zu machen, Nachrichten zu entschlüsseln, die mit MIT PGP verschlüsselt wurden. Diese Änderung wurde in MIT PGP so geschrieben, das sie am 1. September 1994 in Kraft trat. Aufgrund dessen können mit MIT PGP verschlüsselte Nachrichten nicht mehr von 2.4 (und früheren Versionen) gelesen werden. Das Ziel war, daß Leute, die 2.4 und frühere Versionen benutzten, gezwungen wurden, aufzurüsten und somit die patentrechtlich zweifelhafte Version nicht mehr benutzt würde.

Der beste Ausweg ist, auf eine neuere Version aufzurüsten. Wenn Du eine andere Version als MIT PGP benutzt, suche die "legal_kludge"-Option in Deiner Dokumentation: Du solltest in der Lage sein, Deine Ausführung von PGP so zu konfigurieren, daß Nachrichten "im alten Stil" erzeugt werden. In 2.6.2i und 2.6.3i geht das, indem die Zeile "Legal_Kludge=off" in Deiner config.txt-Datei für PGP

eingefügt wird. Beachte, daß "alte" Chiffretexte von den neueren Versionen korrekt gelesen werden können, so daß Du in Korrespondenz mit MIT und 2.3-Benutzern bei ausgeschalteter "Legal_Kludge-" Option am Besten dran bist.

2.2 Wieso beklagt sich PGP so oft über Signaturen?

Version 2.3a führte die "pkcs-compat-" Option ein, die es erlaubte, die Signaturen leicht zu modifizieren, um sie an Industrie-Standards anzupassen. MIT PGP versteht aufgrund der benutzten RSAREF-Bibliothek das alte Signaturformat nicht, ignoriert die Signaturen und gibt eine entsprechende Meldung aus. Das Problem taucht meistens im Zusammenhang mit älteren Schlüsselzertifikaten auf. Sollte Dein Schlüssel solche Signaturen enthalten, bringe die Leute, die Deinen Schlüssel unterschrieben haben dazu, das Gleiche mit einer neuen Version von PGP zu wiederholen. Sollte es trotzdem extrem wichtig sein, ein altes Zertifikat zu überprüfen, besorge Dir eine Version von PGP, die RSA anstelle von RSAREF benutzt, z.B. ViaCrypt.

2.3 Wieso dauert das Verschlüsseln und Entschlüsseln von Nachrichten so lange?

Das Problem kann entstehen, wenn Du sämtliche Schlüssel eines Key-Servers in der Datei pubring.pgp untergebracht hast. PGP muß dann wohl unter einigen Tausenden nach dem gewünschten Schlüssel suchen. Der Ausweg aus diesem Dilemma besteht darin, zwei verschiedene öffentliche Schlüsselbunde zu verwenden (siehe Frage [180]4.2) Der erste Ring, die normale Datei pubring.pgp, sollte nur Empfängern vorbehalten sein, denen Du ziemlich oft Nachrichten schickst. Der zweite Schlüsselbund kann alle anderen Schlüssel enthalten. Du mußt natürlich bei jedem "Kryptiervorhang" den zu benutzenden Schlüsselbund angeben. Von da an wird die Ver-/Entschlüsselung mit Empfängerschlüsseln aus Deinem kürzeren Schlüsselbund viel schneller von statten gehen. Die Ver- und Entschlüsselungszeiten vergrößern sich auch mit der Größe der einzelnen Schlüssel. Ein 2048-Bit Schlüssel wird viel langsamer arbeiten als beispielsweise ein 512-Bit Schlüssel.

2.4 Wie erstelle ich eine zweite Schlüsseldatei?

Laß uns zunächst davon ausgehen, daß Du den gesamten Mammut-Schlüsselbund als Standard in der Datei pubring.pgp untergebracht hast. Du wirst alle häufiger benutzten Schlüssel in separate Schlüsseldateien extrahieren müssen, indem Du das -kx - Kommando benutzt. Dann gib pubring.pgp einen beliebigen neuen Namen. Wir nehmen dafür den Namen "pubring.big" an. Jetzt füge jeden einzelnen der vorher extrahierten Schlüssel in eine neue Datei pubring.pgp ein. Das Kommando dazu ist -ka. Zum Verschlüsseln für einen Empfänger im kurzen Standard-Schlüsselbund benutze das Kommando "pgp -e <Datei> <Empfänger>". Für jemanden im langen Schlüsselbund verschlüsselst Du mit "pgp -e +pubring=c:\pgp\pubring.big <Datei> <Empfänger>". Beachte, daß Du den gesamten Pfad und den Dateinamen des zweiten Schlüsselbundes angeben mußt. Er wird nicht gefunden, wenn Du nur den Dateinamen nennst.

2.5 Wie geht PGP mit Mehrfachadressierungen um?

Beim Verschlüsseln einer Nachricht für mehrere Adressaten wirst Du feststellen, daß die verschlüsselte Datei pro Adressat nur geringfügig länger wird. Der Grund dafür liegt darin, daß der Nachrichtenblock nur ein einziges Mal mit einem zufälligen Schlüssel und IDEA verschlüsselt wird. Es ist dann bloß erforderlich, diesen Schlüssel für jeden Adressaten einmal zu verschlüsseln und ihn im Header der Nachricht unterzubringen. Die komplette Nachricht wächst also um die Größe des

Headers für jede zusätzliche Adresse. (Jedesmal wenn er RSA-verschlüsselt wird, wird der IDEA-Schlüssel um verschiedene Zufallsdaten ergänzt. Dies zur Vermeidung einer bekannten Schwäche von RSA beim Verschlüsseln der gleichen Nachricht an mehrere Empfänger.)

2.6 Wo bekomme ich Skripte, die PGP in meine Email oder News-Programme einbinden?

Es sind viele Skripte und Programme erhältlich, um PGP leichter anwenden zu können. Einen Index findest Du auf [181]http://www.primenet.com/~shauert/.

Wenn Du von Oberflächen, Skripten oder Front-Ends weißt, die hier nicht erwähnt werden, übermittle die URL (oder sonstwie brauchbare Informationen darüber) an den Autor dieser Seite (Scott Hauert, [182]shauert@primenet.com), nicht an mich.

2.7 Wie entschlüssele ich Nachrichten, die ich für Andere verschlüsselt habe?

Bei konventioneller Verschlüsselung kannst Du die Nachricht lesen, indem Du PGP auf sie anwendest und das Paßwort eingibst, das Du beim Verschlüsseln gewählt hast.

Bei der Verschlüsselung mit PGP und öffentlichen Schlüsseln ist das unmöglich, es sei denn, Du hast gleichzeitig an Dich selbst verschlüsselt.

Es gibt eine undokumentierte Einstellung: "EncryptToSelf", die Du in Deiner Konfigurationsdatei "Config.txt" oder in der Kommandozeile auf "on" stellen kannst, wenn Du möchtest, daß PGP Nachrichten immer auch an Dich selbst verschlüsselt. Sei jedoch gewarnt: Würde Dein geheimer Schlüssel kompromittiert (einem Unberechtigten zugänglich), würde diese Einstellung bedeuten, daß sowohl Deine versandten als auch Deine empfangenen Nachrichten zu lesen wären.

2.8 Warum kann ich mit PGP unter Unix keinen Schlüssel generieren?

Das liegt wahrscheinlich daran, daß PGP keine Dateien für die öffentlichen und geheimen Schlüsselbunde erstellen kann. Sollte die Umgebungsvariable PGPPATH nicht definiert sein, wird PGP versuchen, die Dateien ins Unterverzeichnis ".pgp" Deines Hauptverzeichnisses zu legen. Es wird nicht versuchen, das Verzeichnis bei Bedarf einzurichten; statt dessen, wenn es nicht schon existiert, wird PGP nach der Schlüsselerzeugung zusammenbrechen. Das passiert auch, wenn PGPPATH auf ein Verzeichnis verweist, in dem Du über keine Schreibberechtigung verfügst.

Es gibt zwei Auswege: Setze die Umgebungsvariable PGPPATH so, daß sie auf das Verzeichnis mit Deinen Schlüsselringen verweist oder starte "mkdir $HOME/pgp; chmod 700 $HOME/.pgp" bevor Du Deine Schlüssel erzeugst.

2.9 Wenn ich ein Dokument mit einer "Klartextunterschrift" versehe, stellt PGP einigen meiner Zeilen Bindestriche voran. Wozu ist das gut?

PGP tut das wegen des "-----BEGIN PGP MESSAGE-----" (und ähnlicher) Header, den es benutzt, um den Anfang von PGP-Nachrichten zu markieren. Um nichts durcheinander zu bringen, wird "- " jeder Zeile vorangestellt, die mit einem Bindestrich beginnt. Der Extra-Bindestrich und die Leerstelle werden wieder entfernt, wenn Du die Signatur überprüfst. Statt dessen wird der Originaltext wieder hergestellt.

Das Gleiche geschieht mit einigen Zeilen, die spezielle Formeln enthalten, so zum Beispiel "From", weil diese Zeilen andernfalls von Mail-Programmen ausgelassen würden, was wiederum die Signatur verfälschen würde.

2.10 Wie verschlüssele ich mehrere Dateien auf einmal?

PGP akzeptiert zur Verschlüsselung normalerweise nur eine Datei in der Kommandozeile. Einige Betriebssysteme erlauben es Dir, Programme in einer "Batch-" Sequenz aufzurufen. Du kannst diesen Umstand ausnutzen, um PGP automatisch auf mehrere Dateien anzuwenden.

Unter MS-DOS und OS/2 geht das folgendermaßen:
```
for %a in (*.*) do pgp -ea %a userid
```

Du kannst auf diese Weise auch konventionell verschlüsseln, indem Du die undokumentierte "-z -"Option benutzt, um ein Paßwort (oder einen längeren Paßphrase) für die Verschlüsselung aller Dateien festzulegen:
```
for %a in (*.*) do pgp -c %a -z"das Paßwort"
```

In UNIX würde das so aussehen:
```
for a in *; do pgp -ea $a userid; done
```

Einige Oberflächen und "Front-Ends" ermöglichen Dir natürlich ebenfalls, mehrere Dateien auf einmal zu verschlüsseln.

2.11 Wie übergebe ich meinen "Paßphrase" automatisch an PGP?

Drei Möglichkeiten: Am einfachsten ist es wohl, die Umgebungsvariable PGPPASS so zu setzen, daß sie Dein Paßwort ("Paßphrase") enthält. Unter DOS könntest Du das mit der Eingabe "set PGPPASS=Mein geheimer Paßphrase" erreichen.

Das ist ein sehr unsicheres Verfahren, da jeder, der Zugang zu Deinem System hat, Deinen Paßphrase sehen kann. Das schließt alle Leute ein, die während Deiner Mittagspause daherkommen und am DOS-PROMPT Deines Computers "set" eingeben. In einigen UNIX-Versionen ist es ebenfalls möglich, die Umgebungsvariablen eines Anderen zu untersuchen.

Einen weiteren Ansatz, besonders gut von grafischen Oberflächen aus zu nutzen, bietet die "z-" Option. Du mußt -z"Mein Paßphrase" der PGP-Kommandozeile zufügen. Setze den Paßphrase in Anführungszeichen, wenn er irgendwelche Sonderzeichen enthält, wie < oder >, die das Programm irreführen könnten.

Das wiederum ist noch unsicherer auf einem Mehrbenutzersystem. Jeder sieht, welche Programme Du benutzt, inklusive aller Optionen, die Du nutzt.

Die beste, aber auch komplizierteste Methode besteht darin, die Umgebungsvariable PGPPASSFD zu benutzen. Sie sollte eine "Dateibeschreibungs-Nummer" enthalten, die auf die Datei mit Deinem Paßphrase hindeutet. So wird Dein Paßphrase vor jedem außer dem Superuser geschützt, vorausgesetzt, Du hast die Zugriffsrechte richtig festgelegt.

Dank an Jack Gostl <[183]gost.@argos.argoscomp.com> für das Folgende:

> Man findet Informationen darüber im Anhangtext des PGP262-Paketes. Wenn Du PGPPASSFD auf 0 setzt, wird PGP den Paßphrase beim Start vom Standard-Input (Keyboard) lesen.
>
> ```
> PGPPASSFD=0; export PGPPASSFD
> ```

```
echo "PaßPhraseHier" ¦ pgp -east datei empfänger1 empfänger2
```

Patrick J. LoPresti <[184]patl@lcs.mit.edu> fügt hinzu:

> Du kannst auch Eingabeumleitung benutzen, um den Paßphrase aus einer beliebigen Datei zu holen. Der exakte Befehl hängt von der benutzten Oberfläche ab. KSH und ähnliche benutzen "export PGPPASSFD=3", CSH und Verwandte dagegen "setenv PGPPASSFD 3".
>
> ```
> setenv PGPPASSFD 3; pgp -eat file recipient 3 <
> /meine/Paßphrase/Datei
> ```

Das letzte Beispiel bietet den zusätzlichen Vorteil, daß der Standard-Input dem Benutzer weiterhin zur Verfügung steht, um beispielsweise JA/NEIN-Fragen zu beantworten.

2.12 Kann es sein, daß "randseed.bin" von einem Virus infiziert wird?

Die Datei 'Randseed.bin' wird von PGP benutzt, um jedesmal, wenn Du etwas verschlüsselst, einen neuen, zufälligen Schlüssel zu erzeugen. Danach wird sie mit neuen Zufallsdaten aufgefüllt. Ein Virus-Checker wird dann natürlich Veränderungen an der Datei feststellen. Da die Datei "bin" als Erweiterung trägt, glauben die meisten Prüfprogramme, es mit einer ausführbaren Datei zu tun zu haben und werden Dich über einen möglichen Virusfund in Kenntnis setzen.

Dennoch ist diese Datei nur Quelle einiger Zufallswerte und wird niemals "ausgeführt". Es ist daher sicher, sie in die Ausschlußliste Deines Virusscanners aufzunehmen, damit sie in Zukunft nicht mehr beachtet wird. Alternativ kann bei 2.6-Versionen die Zeile Randseed=C:\pgp\random.src in Config.txt aufgenommen werden. Das bringt PGP dazu, die Zufallsbits in dieser Datei statt in randseed.bin zu speichern.

Es wird keinen Schaden anrichten, wenn Du Randseed.bin löschst; PGP wird Dich lediglich um einige zufällige Tastaturanschläge bitten und die Datei beim nächsten Kryptiervorgang neu erstellen.

2.13 Wieso findet MacPGP meinen privaten Schlüssel nicht?

Zbigniew fiedorowicz <[185]fiedorow@math.ohio-state.edu> erklaert:

> Das ist ein echter Fehler in MIT MacPGP 2.6.2 und sicherlich eine FAQ in den PGP-Newsgroups. MIT Mac PGP 2.6.2 erklärt seltsamerweise, daß es Deinen geheimen Schlüssel nicht findet, obwohl es Deinen geheimen Schlüsselbund findet. Das kann gelegentlich passieren. Der Grund dafür ist ein nicht initialisierter "pointer", der eigentlich auf Deine User-ID deuten soll, wenn Du eine festgelegt hast, andernfalls auf den leeren String.
>
> Leider wird er im letzteren Fall nicht initiallisiert und weist auf irgendeinen zufälligen Speicherbereich. Wenn dieser Bereich mit einem Null-Byte startet, wird alles normal ablaufen, und MacPGP wird den ersten geheimen Schlüssel in secring.pgp benutzen. Anderenfalls aber wird MIT MacPGP annehmen, daß Dein User-ID irgend ein zufälliger Abfall ist und konsequenterweise nicht fähig sein, Deinen geheimen Schlüssel zu finden. Umgehen läßt sich das Problem, indem Du Deine Datei Config.txt editierst und die Zeile "MyName="Name_im_geheimen_Schlüssel" einfügst.

2.14 Wie setze ich die TZ-Variable?

Die Umgebungsvariable TZ wird gebraucht, um die Zeitzone, in der sich Dein Computer befindet, einzustellen. Das erlaubt es PGP, Zeitstempel für UTC-Zeiten (früher GMT) zu erzeugen, so daß es keine Widersprüche gibt, wenn jemand in einer anderen Zeitzone die Signatur nachprüft und herausfindet, daß sie in der Zukunft erstellt worden ist oder zu einem anderen unrichtigen Zeitpunkt. Sie wird in Deiner AUTOEXEC.BAT (in DOS) oder CONFIG.SYS (OS/2) definiert. Für andere Systeme, konsultiere das Handbuch.

In den meisten Fällen, kannst Du die Einstellungen aus der folgenden Liste übernehmen:

* Für Los Angeles: SET TZ=PST8PDT
* Für Denver: SET TZ=MST7MDT
* Für Arizona: SET TZ=MST7 (Arizona wendet niemals Sommerzeit an)
* Für Chicago: SET TZ=CST6CDT
* Für New York: SET TZ=EST5EDT
* Für London: SET TZ=GMT0BST
* Für Amsterdam: SET TZ=MET-1DST
* Für Moskau: SET TZ=MSK-3MSD
* Für Aukland: SET TZ=NZT-12DST

Für andere Länder muß die komplette Darstellung der TZ-Variable benutzt werden. Ausführlicher heißt das: SET TZ=SSS[+|-]nDDD,sm,sw,sd,st,em,ew,ed,et,shift

Wobei "SSS", "n" und DDD" die Werte der vereinfachten Darstellung sind. In der langen Form müssen alle anderen Werte wie folgt spezifiziert werden.

"sm"
: ist der Anfangsmonat (1-12) der Sommerzeit

"sw"
: ist die Anfangswoche (1-4, vom Begin gezählt, oder -1 bis -4, vom Ende). 0 zeigt an, daß ein bestimmter Tag des Monats angegeben werden muß.

"sd"
: ist der Anfangstag (0 bis 6 [wobei 0 der Sonntag ist] wenn "sw" ungleich 0 ist, oder 1 bis 31, wenn "sw" gleich 0 ist)

"st"
: ist die Startzeit in Sekunden seit Mitternacht (also 3600 für 01:00 Uhr)

"em", "ew", "ed" und "et"
: definieren das Ende der Sommerzeit und nehmen die gleichen Werte an.

"shift"
: ist der Zeitunterschied bei der Umstellung auf Sommerzeit, in Sekunden (also 3600 wenn während der Sommerzeit 1 Stunde addiert werden muß).

Zum Beispiel wurde für Großbritannien 1995 die Einstellung SET TZ=GMT0BST,3,0,26,3600,10,0,22,3600,3600 erwartet (26-März-01Uhr-1Std vorstellen-22-Oktober-1Std-zurückstellen).

2.15 Wie erkenne ich, ob das PGP-Kommando korrekt ausgeführt worden ist?

Normalerweise läuft PGP im "interaktiven" Modus, und somit kannst Du immer auf dem Bildschirm lesen, was schief gegangen ist, wo und hoffentlich warum. Aber, solltest Du PGP in einer Batch-Datei benutzen

wollen oder im Hintergrund, mußt Du auf andere Weise herausfinden ob die Aktion erfolgreich war. Der gewöhnlich eingeschlagene Weg dazu, ist der Gebrauch des "Beendigungscodes", den PGP zurueckgibt.

Um zu sehen, ob PGP ausführen kann, was Du verlangst, mußt Du die "+batchmode" -Option in der Kommandozeile angeben (Um zu vermeiden, daß Du an Eingabeaufforderungen festhängst, die Dich um "Ja" oder "Nein" bitten, nimm die "+force" -Option hinzu). PGP wird dann 0 zurückgeben, wenn alles in Ordnung ist und 1, wenn etwas schief gelaufen ist.

Der PGP Quellcode enthält eine Liste von Beendigungscodes, deren Ausgabe erwartet wird, wenn die damit verbundenen Ereignisse eintreten. Es scheint dies nicht immer so zu funktionieren, wie erwartet. Zum Beispiel gibt PGP den Beendigungscode 31 zurück, wenn kein Paßphrase festgelegt worden ist um die Datei zu entschlüsseln, aber wenn Du versuchst eine Signatur zu prüfen, wird Beendigungscode 1 benutzt, um jedweden Fehler anzuzeigen, einschließlich "Kein Schlüssel um die Signatur zu prüfen" und "Keine korrekte Signatur".

Warum fragt PGP 5.0 nicht mehr nach zufälligen Tastendrücken?

Die Tastendrücke waren notwendig, um zufällige Ereignisse zu bekommen. PGP 5.0 benutzt Systemereignisse, die die ganze Zeit stattfinden, um ständig die Datei randseed.bin zu füllen. Diese Ereignisse umfassen Festplattenzugriffe, Tastatureingaben, Mausbewegungen und andere Dinge, die zufällig genug sind. Zur Überprüfung kannst Du beobachten, wie randseed.bin auch dann verändert wird, wenn Du nicht PGP benutzt.

Sind PGP 5.0/5.5 und PGP 2.6.x kompatibel?

PGP 5.x ist abwärtskompatibel zu PGP 2.6.x. Das bedeutet, PGP 5.x kann mit den Ausgaben von PGP 2.6.x problemlos arbeiten. Werden nur RSA Schlüssel mit IDEA und MD5 verwendet, so sind auch umgekehrt die Ausgaben von PGP 5.x mit PGP 2.6.x verarbeitbar. Ein wenig problematisch sind ElGamal oder DSS Zertifikate von PGP 5.x unter RSA Schlüsseln. Diese Zertifikate werden von PGP 2.6.x beim Kommando '-kc' fehlerhaft verarbeitet. Die Version PGP 2.6.3in behebt diesen (unwesentlichen) Fehler.

PGP 2.6.x arbeitet mit PGP 5.x zusammen, wenn ausschließlich MD5, RSA und IDEA verwendet werden.

Sicherheitsfragen

3.1 Wie sicher ist PGP?

Die große Unbekannte in jedem Verschüsselungs-Konzept, das auf RSA basiert, ist: Gibt es einen effizienten Weg zur Faktorisierung großer Zahlen, oder: Gibt es eine Art "Hintertür-Algorithmus", der den Code ohne das Faktorisierungsproblem brechen kann? Sogar, wenn kein solcher Algorithmus existiert, glaubt man immernoch, daß RSA das schwächste Glied in der PGP-Kette ist.

Alle möglichen Angriffe gegen-, oder mögliche Fehler in PGP zu besprechen, würde den Rahmen dieser FAQ sprengen. Wenn Du mehr erfahren willst, als hier angeboten wird, schau nach in "infiNity's PGP Attack FAQ":
[208]http://www.stack.nl/~galactus/remailers/attack-faq.html

3.2 Kann man PGP brechen, indem man sämtliche, denkbaren Schlüssel ausprobiert?

Das ist eine der ersten Fragen, die Leute stellen, wenn sie zum ersten Mal in die Kryptographie eingeführt werden. Sie verstehen das Ausmaß des Problems nicht. Für das IDEA-Verschlüsselungskonzept ist ein 128-Bit-Schlüssel erforderlich. Jede der 2^128 möglichen Kombinationen wäre als Schlüssel zulässig, und nur der eine Schlüssel würde erfolgreich alle Nachrichtenblöcke entschlüsseln. Laß uns annehmen, daß Du einen Spezialchip entwickelt hättest, der eine Milliarde Schlüssel pro Sekunde ausprobieren könnte. Das ist _weit_ jenseits der heutigen Möglichkeiten. Laß uns ebenfalls sagen, Du könntest es Dir leisten, eine Milliarde solcher Chips zur gleichen Zeit, auf das Problem anzusetzen. Es würde immernoch 10.000.000.000.000 Jahre dauern, all die möglichen 128-Bit-Schlüssel auszuprobieren. Das ist in etwa tausendmal das Alter des bekannten Universums! Obwohl die Geschwindigkeit der Computer weiterhin steigt und ihre Kosten sehr schnell sinken, wird es wahrscheinlich niemals soweit kommen, daß IDEA durch diese "Brute Force Attacke" (etwa: 'Brachialangriff') geknackt werden könnte.

Die einzige Form des Angriffs, die erfolgreich sein könnte, ist eine, die das Problem von einem mathematischen Standpunkt aus löst: Durch das Analysieren der Umformungen, die zwischen Klartextblöcken und ihren Chiffretext-Äquivalenten stattfinden. IDEA ist immernoch ein ziemlich neuer Algorithmus, und es muß noch viel Arbeit daran verrichtet werden, weil er mit der Theorie der komplexen Zahlen in Verbindung steht, aber bislang scheint es keinen Algorithmus zu geben, der sehr viel besser geeignet wäre, eine IDEA-Verschlüsselung

aufzulösen als die brute force Attacke, die wir schon als unpraktikabel dargestellt haben. Die nichtlineare Transformation, die in IDEA stattfindet, stuft es in eine Klasse extrem schwierig zu lösender Probleme ein.

3.3 Wie sicher ist die konventionelle Verschlüsselungs-Option (-c)?

Unter der Annahme, daß Du einen guten, starken, zufälligen Paßphrase benutzt, ist sie viel stärker als der normale Verschlüsselungsmodus, weil Du RSA ausläßt, das als das schwächste Glied in der Kette angesehen wird. Natürlich wirst Du in diesem Modus vorab geheime Schlüssel mit jedem der Empfänger austauschen müssen, indem Du eine andere sichere Methode der Kommunikation benutzt, wie zum Beispiel Treffen unter vier Augen oder einen vertrauenswürdigen Kurier.

Diese Option ist besonders nützlich, wenn Du sensible Dateien sichern oder verschlüsselte Dateien zu einem anderen System bringen willst, wo Du sie entschlüsseln wirst. Nun mußt Du nicht Deinen geheimen Schlüssel mitnehmen.

Das wird auch nützlich sein, wenn Du Deinen geheimen Schlüssel verloren hast. Und Du kannst Dir für jede Datei, die Du verschlüsselst, einen anderen Paßphrase aussuchen, so daß ein Angreifer, der es schafft, eine Datei zu entschlüsseln, jetzt nicht auch all die anderen Dateien entziffern kann.

3.4 Kann die NSA (amerikanischer Geheimdienst "National Security Agency") RSA brechen?

Diese Frage ist viele Male gestellt worden. Wäre die NSA fähig, RSA zu brechen, würdest Du von ihr darüber wahrscheinlich niemals etwas hören. Jetzt, wo RSA immer populärer wird, wäre es ein sehr gut gehütetes Geheimnis. Das beste Argument dagegen ist die Tatsache, daß der Algorithmus für RSA weltweit bekannt ist. Es gibt viele gute Mathematiker, und es wäre schwierig, solch eine Entdeckung zu verstecken.

Aus diesem Grunde, wenn Du im USENET Nachrichten liest, die behaupten, daß "jemand ihnen gesagt habe", die NSA sei fähig, PGP zu brechen, glaube es nicht ohne weiteres, und frage nach schriftlichen Unterlagen darüber, von wo die Information stammt. Speziell die Meldung auf [209]http://www.quadralay.com/www/Crypt/NSA/break-pgp.html ist ein Witz.

3.5 Wurde RSA jemals öffentlich gebrochen? Was ist RSA-129?

Zwei RSA-verschlüsselte Nachrichten sind öffentlich geknackt worden.

Zunächst ist da der RSA-129-Schlüssel. Die Erfinder von RSA haben eine Meldung veröffentlicht, die mit einem 129-stelligen (430 Bits), oeffentlichen Schlüssel verschlüsselt worden ist und boten der ersten Person, die die Nachricht entschlüsseln koennte, 100$. 1994 faktorisierte ein internationales Team, das von Paul Leyland, Derek Atkins, Arjen Lenstra und Michael Graff koordiniert wurde, erfolgreich diesen öffentlichen Schlüssel und stellte den Originaltext wieder her. Die Nachricht lautete: _THE MAGIC WORDS ARE SQUEMISH OSSIFRAGE_

Sie führten eine riesige Freiwilligenaktion an, während derer die Arbeit über E-mail, Fax und reguläre Post an Teilnehmer im Internet verteilt wurde, die ihren Teil verarbeiteten und die Resultate zurückschickten. Etwa 1600 Maschinen nahmen Teil, mit Rechnerleistungen, die vom Faxgerät bis zum Cray Supercomputer reichten. Sie benutzten den damals besten bekannten

Faktorisierungsalgorithmus; seither sind bessere Methoden entdeckt worden, aber das Resultat ist immernoch lehrreich hinsichtlich des Arbeitsaufwandes, der benötigt wird, eine RSA-verschlüsselte Nachricht zu brechen.

Die Koordinatoren haben geschätzt, daß das Projekt etwa 8 Monate realer Zeit und nahezu 5000 MIPS-Jahre Rechenzeit gebraucht hat.

Was hat all das mit PGP zu tun? Der RSA-129-Schlüssel ist, was die Sicherheit angeht, etwa dem 426-Bit PGP-Schlüssel ebenbürtig. Daß der leicht zu brechen ist, wurde bei diesem Projekt gezeigt. PGP pflegte 384-Bit-Schlüssel als "casual grade" Sicherheitsstufe zu empfehlen; jüngere Versionen bieten 512 Bits als empfohlene Mindestsicherheitsstufe an.

Beachte, daß die Aktion nur einen einzigen RSA-Schlüssel brach. Nichts wurde im Verlauf des Experimentes entdeckt, das irgendeinen anderen Schlüssel weniger sicher macht, als er es vorher gewesen ist.

Ein Jahr später wurde der erste wirkliche PGP-Schlüssel geknackt. Es war der berühmte _Blacknet_ Schlüssel, ein 384-Bit-Schlüssel der anonymen Existenz, die als "Blacknet" bekannt war. Ein Team, das aus Alec Muffet, Paul Leyland, Arjen Lenstra und Jim Gillogly bestand, schaffte es, genügend Rechenleistung zusammenzubringen (annähernd 1300 MIPS), um den Schlüssel in drei Monaten zu faktorisieren. Er wurde dann benutzt, um eine öffentlich zugängliche Nachricht, die damit verschlüsselt worden war, zu entschlüsseln.

Das Allerwichtigste bei dieser Attacke war, daß sie unter fast kompletter Geheimhaltung stattfand. Anders als beim RSA-129-Angriff, gab es keine Publicity um dieses Projekt, bis es vollendet war. Die meisten Computer arbeiteten nur in der Freizeit daran, und die gesamte Leistung liegt sehr wohl im Bereich der Möglichkeiten einer großen, vielleicht sogar mittelgroßen, Organisation.

3.6 Wie sicher ist die "Nur zur Ansicht"- Option(-m)?

Sie ist überhaupt nicht sicher. Es gibt viele Wege, sie zu überwinden. Der Einfachste ist wahrscheinlich schlicht, Deine Bildschirmausgabe in eine Datei umzuleiten, wie folgt: "pgp [Dateiname] > [neuer Dateiname]"

Die "-m"-Option war nicht als idiotensichere Möglichkeit gedacht, um die Erzeugung von Klartexten zu verhindern, sondern einfach als Warnung an die Person, die die Datei entschlüsselt, daß sie besser keine Kopie des Klartextes auf ihrem System behalten sollte.

3.7 Was ist, wenn ich meinen Paßphrase vergesse?

Kurz: Vergiß ihn nicht. Wenn Du Deinen Paßphrase vergißt, gibt es absolut keinen Weg, irgend eine verschlüsselte Datei zurückzugewinnen. Wenn Du besorgt bist, Deinen Paßphrase zu vergessen, könntest Du eine Kopie Deines geheimen Schlüssels anfertigen, dann den Paßphrase in einen anderen abändern, und schließlich den geheimen Schlüssel mit dem geänderten Paßphrase an einem sicheren Ort aufbewahren.

3.8 Warum wird der Begriff "Paßphrase" (oder Mantra) anstelle "Paßword" benutzt?

Das geschieht, weil die meisten Leute, wenn sie gebeten werden, ein Paßwort zu wählen, ein einfaches gebräuchliches Wort aussuchen. Das kann von einem Programm erraten werden, das ein Wörterbuch benutzt um Paßwörter an einem System auszuprobieren.

Weil die meisten Menschen tatsächlich kein wirklich zufälliges Paßwort auswählen wollen, bei dem die Buchstaben und Ziffern in einem unsinnigen Muster gemischt sind, wird der Ausdruck "Paßphrase" benutzt, um die Leute zu drängen, wenigstens mehrere beziehungslose Worte hintereinander als Paßphrase zu verwenden.

3.9 Was ist der beste Weg um PGP zu knacken?

Zur Zeit ist der beste Angriff gegen PGP selbst eine Wörterbuchattacke gegen den Paßphrase. Dabei entnimmt ein Programm einem Wörterbuch Worte und reiht sie, im Bestreben Deinen Paßphrase zu erraten, auf verschiedene Weise hintereinander.

Deshalb ist es so wichtig, einen starken Paßphrase zu wählen. Viele dieser "Cracker-"Programme sind sehr raffiniert und machen sich Dialekte zu Nutze, populäre Aussprüche und grammatikalische Regeln, um ihre Versuche zusammenzusetzen. "Ein-Wort"-Paßphrases, Eigennamen (speziell berühmte) oder bekannte Zitate sind nahezu immer von einem Programm zu knacken, das überhaupt irgendwelche "Feinheiten" besitzt.

Es ist ein Programm erhältlich, das konventionell verschlüsselte Nachrichten durch Raten der Paßphrases knacken kann. Es wendet keinerlei Kryptoanalyse an, so daß Deine Dateien immernoch sicher sein werden, wenn Du einen starken Paßphrase wählst. Siehe [210]http://www.voicenet.com~/markm/pgpcrack.html für mehr Informationen und das Programm selbst.

Es gibt auch andere Methoden, um an den Inhalt einer verschlüsselten Nachricht zu gelangen, wie Bestechung, Schnüffelei nach der elektronischen Abstrahlung des Computers, der die Nachrichten verarbeitet (oft als "TEMPEST-"Attacke bezeichnet), Erpressung oder "rubber-hose-"Kryptographie- solange mit einem Gummischlauch auf den Kopf schlagen, bis Du den Paßphrase preisgibst.

3.10 Können meine Nachrichten gelesen werden, wenn mein privater Schlüssel gestohlen wurde?

Nein, solange sie nicht auch Deinen geheimen Paßphrase gestohlen haben, oder Dein Paßphrase anfällig für eine brute-force-Attacke ist. Kein Teil ist ohne den anderen brauchbar. Du solltest trotzdem diesen Schlüssel zurückziehen und einen frischen mit einem anderen Paßphrase generieren. Bevor Du Deinen alten Schlüssel zurückziehst, willst Du vielleicht eine neue User-ID anhängen, die feststellt, welche Deine neue Schlüssel-ID ist, so daß Andere von Deiner neuen Adresse erfahren.

3.11 Wie wähle ich meinen Paßphrase?

Die ganze Sicherheit, die in PGP zur Verfügung steht, kann absolut nutzlos gemacht werden, wenn Du keinen guten Paßphrase wählst, um Deinen geheimen Schlüsselbund zu verschlüsseln. Zu viele Leute benutzen ihren Geburtstag, ihre Telefonnummer, den Namen einer/eines Geliebten oder irgendein leicht zu ratendes, gebräuchliches Wort.

Während es eine Anzahl von Vorschlägen zur Erzeugung guter Paßphrases gibt, erhält man das Optimum an Sicherheit, wenn die Zeichen des Paßphrases komplett zufällig ausgesucht werden. Er kann etwas schwieriger zu merken sein, aber die zusätzliche Sicherheit ist es wert. Als absolut minimaler Paßphrase, würde ich eine zufällige Kombination von mindestens 8 Buchstaben und Ziffern vorschlagen, wobei 12 die bessere Wahl sind. Mit einem 12-Buchstaben-Paßphrase, der aus den Kleinbuchstaben von a-z besteht und den Ziffen 0-9, erhälst Du