Pretty Good Privacy*

Rainer W. Gerling, Fürstenfeldbruck

1 Die Geschichte von Pretty Good Privacy

Der US Senat startete 1991 ein Gesetzgebungsverfahren[1], in dem verlangt wurde, dass jedes Verschlüsselungsprodukt eine Hintertür für staatliche Stellen haben muss. Unter dem Eindruck dieses Gesetzentwurfes begann Philipp R. Zimmermann, PGP zu schreiben. Sein Ziel war es, Kryptographie weit zu verbreiten (Cryptography for the masses), um dadurch das Verbot der Kryptographie zu erschweren.

PGP 1.0 verwendete bereits das RSA Verfahren. Zur symmetrischen Verschlüsselung verwendete P.R. Zimmermann einen selbst entworfenen Algorithmus namens Base-O-Matic. Wegen kryptographischer Schwächen verschwand dieser Algorithmus wieder in der Versenkung. Um verschlüsselte Dateien (die ja Binärdateien sind) per E-Mail übertragen zu können, verwendete PGP 1.0 das Verfahren UUEncode zur Kodierung der Datei mit druckbaren Zeichen. Als Hash-Funktion wurde in PGP 1.0 der MD4 Algorithmus verwendet.

Ab Version 2.0 benutzte PGP dann den IDEA Algorithmus als symmetrischen Verschlüsselungsalgorithmus. Da bei MD4 eine Schwäche entdeckt wurde und Ron Rivest diese mit MD5 behob, verwendete Zimmermann dann auch konsequenterweise diese neue Hash-Funktion.

Ohne Zutun von Zimmermann verbreitete sich PGP um die ganze Welt. Damit kam dann die US-Regierung ins Spiel, die den Export von Verschlüsselungssoftware mit dem Export von Kriegswaffen gleich stellte. Ein mehrjähriges Ermittlungsverfahren wegen des illegalen Exports von Verschlüsselungssoftware gegen Philipp Zimmermann wurde schließlich ohne Anklage zu erheben eingestellt.

Ein weiteres rechtliches Problem im Zusammenhang mit PGP entstand an der Patentfront. Das RSA-Verfahren ist vom MIT für die USA patentiert[2], und eine Firma mit Namen Public Key Partners[3] hat das alleinige Recht zur Vermarktung dieses Patents. Da PGP das RSA Verfahren verwendete, verletzte es die Lizenzrechte von Public Key Partners. In den USA und Europa ist außerdem der IDEA Algorithmus patentiert. Somit ist außerhalb der USA PGP nur für nicht kommerzielle Zwecke kostenlos verwendbar. Für kommerzielle Anwendungen wird eine IDEA Lizenz benötigt. In den USA ist keine legale Anwendung (auch keine nicht-kommerzielle) ohne RSA-Lizenz möglich.

Die Verbreitung der PGP Version 2.3 wurde durch diese rechtlichen Auseinandersetzungen nicht gebremst, ganz im Gegenteil.

Schließlich nimmt sich das MIT als Patentinhaber der Situation an. Philip Zimmermann erstellt eine neue Version von PGP (2.5), die für das RSA-Verfahren anstelle der Zimmermannschen Bibliothek MPILIB die Referenzbibliothek RSAREF von RSA Data Security Inc. verwendet. Die RSAREF Bibliothek ist in den USA für nichtkommerzielle Anwendungen frei verfügbar. Das MIT übernimmt auch die offizielle Verbreitung von PGP in den USA. Diese Version von PGP erreicht schließlich die Versionsnummer 2.6.3.

Für kommerzielle Anwendungen kann in den USA von VIACrypt eine kommerzielle, sauber lizenzierte Version von PGP (erst 2.7 später 4.0 und 4.5) gekauft werden.

Außerhalb der USA ist PGP immer noch nicht legal verwendbar, da die RSAREF Lizenz eine Nutzung außerhalb der USA explizit verbietet. Schließlich erstellt Stale Schuhmacher eine internationale Version PGP 2.6.3i, die wieder die außerhalb der USA legale Bibliothek MPILIB verwendet. Diese Version wird außerhalb der USA verbreitet und gilt quasi als die offizielle internationale PGP Version.

Nachdem das Ermittlungsverfahren gegen Philipp Zimmermann eingestellt ist, gründet er die Firma PGP Inc., kauft ViaCrypt und beginnt die lange erwartete Version „3.0“ zu programmieren. Schließlich erscheint auch 1997 diese Version, aber unter der Versionsnummer 5.0. Es folgt eine Version 5.5. Dann wird PGP Inc. von NAI/McAfee gekauft. Es erscheinen die Versionen 6.0.x und 6.5.x.

Die Versionen ab 5.0.x bieten eine grafische Oberfläche für Verschlüs-

INHALT:

* gekürzte und überarbeite Version des Kapitels „PGP und Schlüsselserver“ aus R.W. Gerling, Verschlüsselung im betrieblichen Einsatz, Datakontext Fachverlag, Frechen 2000.

[1] Senate Bill 266; Dieser Vorschlag wurde nie Gesetz.

[2] U.S. Patent Nr. 4405829, erteilt am 20. September 1983

[3] Diese Firma besteht aus RSA Data Security Inc. und Caro Kahn Inc. der Muttergesellschaft von Cylink Inc. Sie existiert nicht mehr.

selung und Schlüsselmanagement. Außerdem wird eine einfache Integration in gängige E-Mail Programme mitgeliefert.

Bestandteil von PGP ist/war ein Programm zur Verschlüsselung von Dateien in Echtzeit: PGPdisk (in der Freeware Version 6.5.x ist es nicht mehr enthalten). Ab Version 6.5 gehört auch eine VPN Lösung PGPnet zu PGP. Mit dieser VPN Lösung kann der komplette Datenverkehr im Netzwerk zwischen zwei (Windows-) Rechnern verschlüsselt werden.

Für die Microsoft Mail-Programme (Exchange, Outlook und Outlook Express) gibt es PGP Plugins von NAI, ebenso für Eudora und den Lotus Notes Klienten. Für Pegasus Mail für Windows[4] ist mit QDPGP[5] auch ein PGP Plugin erhältlich. Lediglich Netscape Mail erlaubt die Verwendung von PGP Plugins nicht. Hier hilft nur der Umweg über die Zwischenablage mit Programmen wie CryptEdit[6] bzw. die Hotkey-Funktionalität in PGP 6.5.1.

Ein neues Kapitel in der PGP Geschichte beginnt im September 1999, als GNU Privacy Guard 1.0.0 freigegeben wird[7]. Dieses Programm kann etwas lax als eine GNU Version von PGP angesehen werden. Es ist völlig neu programmiert, verwendet keine patentierten Algorithmen und erzeugt OpenPGP kompatible Dateien. In dem Standard OpenPGP[8] wird das Dateiformat in Anlehnung an PGP definiert, damit unabhängige Implementierungen möglich sind. PGP 6.x ist nicht hundertprozentig OpenPGP kompatibel aber GnuPG kennt eine Kommandozeilenoption, die PGP kompatible Dateien erzeugt. Der RSA und der IDEA Algorithmus stehen nur über Plugins[9] zur Verfügung. Mit diesen Plugins ist GnuPG sogar PGP

Abbildung 1: Das Fenster mit den Eigenschaften des DSS Schlüssels.

2.6.3 kompatibel. Damit steht erstmals eine PGP kompatible freie und rechtlich einwandfreie Implementierung zur Verfügung.

GnuPG ist keine grafische Windows Anwendung, sondern ein klassisches Kommandozeilen Programm. Damit erleben wahrscheinlich die schon abgeschriebenen PGP-Windows-Shells einen neuen Frühling.

2 Der eigene PGP Schlüssel

Um einen eigenen PGP Schlüssel zu generieren starten wir den Schlüsselerzeugungsassistenten. Dazu wählen wir im PGP Symbol in dem Task-Bar die Option „PGPkeys Starten“ und dann im Menu den Punkt „Schlüssel -> Neuer Schlüssel ...“. Der Schlüsselerzeugungsassistent führt uns durch die Schlüsselgenerierung. Im zweiten Fenster muss unser Name und unsere E-Mail Adresse eingegeben werden.. Die E-Mail Adresse sollte möglichst lange gültig bleiben, damit der Schlüssel nicht zu oft geändert werden muss. Wer öfters den Internet-Provider wechselt oder mehre Provider hat, der sollte darüber nachdenken, eine E-Mail Adresse bei einem Freemail Provider zu beantragen. In einer Firma sollten möglichst E-Mail Adressen der Form name@firma.de benutzt werden. Die häufig übliche Variante name@mail-server.firma.de macht hier wenig Sinn.

Bei der Schlüsselpaargröße wählen wir die Option 2048 Bit (2048 Diffie-Hellman/1024 DSS) aus. Diese Schlüsselgröße sollte für einige Jahre ausreichen. Nun müssen wir noch die Gültigkeitsdauer des Schlüssels festlegen. Entweder bleibt der Schlüssel für immer gültig, oder wir geben ein Verfalldatum ein. Im Normalfall sollte hier eine Gültigkeitsdauer von

[4] http://www.pegasus.usa.com
[5] http://community.wow.net/grt/qdpgp.html
[6] http://madhatter.skuz.net/agent/util.html
[7] http://www.gnupg.org
[8] OpenPGP RFC 2440
[9] Diese Plugins finden sich unter ftp://ftp.gnupg.org/pub/gcrypt/contrib/rsa.c und ftp://ftp.gnupg.org/pub/gcrypt/contrib/idea.c.

ca. fünf Jahren zugrunde gelegt werden.

Wenn wir eine Firmenversion mit einem voreingestellten firmenweiten Unterschriftsschlüssel (CSK[10]) vor uns haben, werden wir jetzt aufgefordert, diesen CSK zu signieren und ihm damit das Vertrauen auszusprechen[11]. Dies darf nicht geschehen, ohne dass wir den Fingerabdruck dieses Schlüssels vorher überprüft haben. An der entsprechenden Stelle der Schlüsselgenerierung ist das leider nicht möglich. Da aber von der Authentizität des CSK viel abhängt, kann man an dieser Stelle nur die Schlüsselgenerierung verlassen, um den CSK zu überprüfen, wenn man es nicht vorher getan hat.

Als nächstes werden wir nach dem Passwort für unseren privaten Schlüssel gefragt. Hier muss zweimal ein mindestens 8 Zeichen langes Passwort eingegeben werden. Dieses Passwort sollte so lang wie möglich sein und hinreichend gemischt aus Großbuchstaben, Kleinbuchstaben, Ziffern und Sonderzeichen bestehen. Das Passwort schützt unseren privaten Schlüssel und damit so zusagen unsere digitale Identität. Es sollte deshalb entsprechend sorgfältig ausgesucht sein. Nun werden die Schlüsselpaare erzeugt. Die Option „Meinen Schlüssel jetzt an den Root-Server senden“ sollten wir zu diesem Zeitpunkt nicht aktivieren.

Wichtig ist das sofortige Sichern des frisch generierten Schlüsselpaares. Wenn wir PGPkeys nach der Schlüsselerzeugung direkt verlassen, werden wir aufgefordert, unsere gerade erzeugten Schlüssel zu sichern. Hierbei wird der komplette öffentliche und private Schlüsselring gespeichert. Es empfiehlt sich, die beiden Schlüsselringe getrennt auf zwei Disketten zu speichern und diese beiden Disketten in jeweils einen Umschlag zu tun. Es ist auch kein Fehler, zusätzlich einen Zettel mit dem Passwort des privaten Schlüssels in den Umschlag zu geben. Der Umschlag muss aber so verschlossen werden, dass man einen unberechtigten Zugriff auf den Inhalt erkennen kann. Die Umschläge werden gemeinsam mit den Wertsachen aufbewahrt.

Abbildung 2: Das Fenster mit den Eigenschaften des Diffie-Hellman Schlüssels.

Über „Schlüssel -> Schlüsseleigenschaften“ können wir uns jederzeit die Details unseres ersten Schlüssels anzeigen *(Abb. 1)*.

Wir finden in der Anzeige alle wichtigen Informationen über unseren Schlüssel. Die Schlüssel ID und den Fingerabdruck benötigen wir zur Authentisierung der Schlüssel. Diese Information wird typisch an Dritte weitergegeben. So kann man z.B. den Fingerabdruck gut auf die Rückseite der Visitenkarte drucken. Das Kästchen „Unausgesprochenes Vertrauen“ findet man nur, wenn sowohl der private als auch der öffentliche Schlüssel im Schlüsselbund sind. Hier kann auch das Passwort, das den privaten Schlüssel schützt, geändert werden.

Alle Angaben hier beziehen sich auf den nur zum Digitalen Signieren bestimmten DSS Schlüssel. Der Teil des Schlüssels, der sich hinter Diffie-Hellman[12] versteckt, ist über die zweite Seite „Teilschlüssel“ zugänglich *(Abb. 2)*.

Zu einem PGP-Schlüssel können durchaus mehrere Diffie-Hellman Schlüssel gehören. Die Diffie-Hellman Schlüssel werden zum Verschlüsseln der Nachrichten verwendet. Werden alle Diffie-Hellman Schlüssel gelöscht, so kann dieses

[10] Company Signing Key

[11] Die Zeitschrift c’t hat z.B. im Frühjahr 1999 eine PGP Version verteilt, in der der Schlüssel „ct magazine CERIFTCATE <pgpCA@ct.heise.de>“ als CSK eingetragen ist.

[12] In OpenPGP und GnuPG wird das Diffie-Hellman Verfahren korrekt als ElGamal Verfahren bezeichnet.

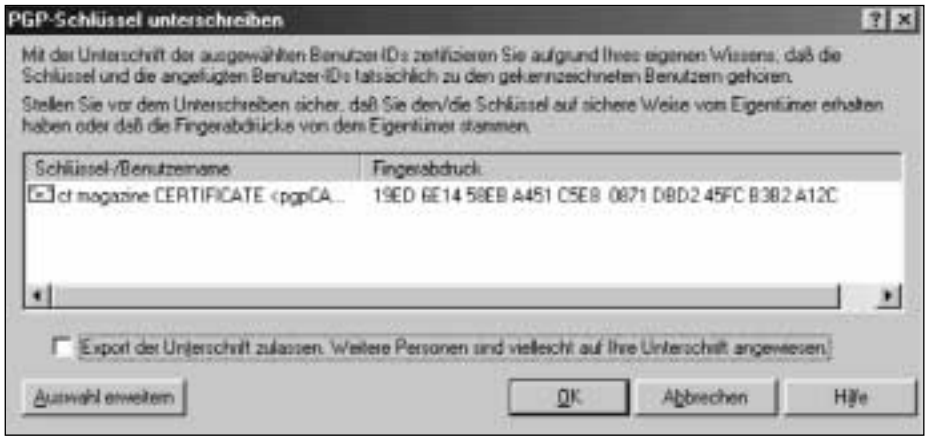

Abbildung 3: Der Dialog zum Signieren anderer Schlüssel.

DSS Schlüsselpaar nicht mehr zum Verschlüsseln benutzt werden, sondern nur noch zum Digitalen Signieren.

Soll ein Schlüsselpaar nur zum Signieren anderer Schlüssel verwendet werden, sollten die/der Diffie-Hellman Schlüssel gelöscht werden.

3 PGP Schlüssel verwalten

PGP speichert Schlüssel in so genannten Schlüsselringen. In dem privaten Schlüsselring (private key ring) sind alle privaten Schlüssel des Benutzers gespeichert. In dem öffentlichen Schlüsselring (public key ring) sind alle öffentlichen (eigene und fremde) Schlüssel, die man regelmäßig benutzt, gespeichert. Der öffentliche Schlüsselring ist die lokale Schlüsseldatenbank des Benutzers.

Zuminderst vor der Version 6.5.x kennt PGP keine Zertifikate. Bei der Generierung der Schlüssel mit PGP wird der öffentliche Schlüssel mit dem eigenen privaten Schlüssel unterschrieben. Damit ist aber gegenüber Dritten keine Fälschungssicherheit gegeben. Bekannte tauschen untereinander ihre Schlüssel aus, verifizieren die Authentizität durch schriftlichen, telefonischen oder persönlichen Austausch des Fingerabdrucks des Schlüssel und signieren einen authentisierten öffentlichen Schlüssel dann selbst. In der Sprache von PGP wird dies als „Web of Trust“ (Netz des Vertrauens) bezeichnet. Durch dieses Verfahren haben PGP Schlüssel häufig sehr viele Signaturen.

Erhält man einen neuen Schlüssel mit mehreren Signaturen, wird automatisch geprüft, ob man einem der Signatur-Schlüssel vertraut. Einem Schlüssel vertrauen heißt, andere Schlüssel, die mit einem solchen vertrauenswürdigen Schlüssel signiert sind, werden als gültig (d.h. echt oder authentisch) akzeptiert.

Das Vertrauen beginnt grundsätzlich mit dem eigenen Schlüssel. Nach der Generierung des eigenen Schlüssels ist dieser gültig und vertauenswürdig. Wenn wir jetzt z.B. den Schlüssel der „`ct magazine CERTIFICATE <pgpCA@ct.heise.de>`“ bekommen, so ist der Schlüssel ungültig und ohne Vertrauen. In jeder c’t steht im Impressum der Fingerabdruck dieses Schlüssels: `19ED 6E14 58EB A451 C5E8 0871 DBD2 45FC B3B2 A12C`. Wenn wir ihn überprüft haben (in einer c’t, nicht auf diesem Artikel), können wir ihn mit unserem eigenen Schlüssel signieren. Hierzu starten wir PGPkeys, markieren den betreffenden Schlüssel und wählen im Menu „Schlüssel -> Unterzeichnen...“ aus *(Abb. 3)*.

Wenn wir den Schlüssel nur für unseren eigene Gebrauch für gültig erklären wollen, so klicken wir auf OK. Dann wird aber unsere Signatur beim Export des Schlüssels nicht mit exportiert. Markieren wir jedoch das Kästchen „Export der Unterschrift zulassen..“ wird unsere Unterschrift mit exportiert.

Durch unsere Signatur ist der Schlüssel gültig geworden. Wenn wir der c’t Zertifizierungsinstanz vertrauen, d.h. wenn wir davon überzeugt sind, dass die c’t Zertifizierungsinstanz nur Schlüssel unterschreibt, nachdem die Identität des Schlüsselinhabers überprüft wurde, dann können wir jetzt das Vertrauen in den Schlüssel einstellen. Dazu wählen im Menu „Schlüssel -> Schlüsseleigenschaften...“ aus.

Dann schieben wir den Regler von „Nicht vertrauenswürdig“ nach „Vertrauenswürdig“. Nach dem Klick auf „Schließen“ sind alle öffentlichen Schlüssel, die von der c’t Zertifizierungsinstanz unterschrieben wurden, gültig. Wenn wir weiteren Zertifizierungsinstanzen trauen, können wir diesen auch das Vertrauen aussprechen. Das Vertauen kann nur einem gültigen Schlüssel ausgesprochen werden.

Nochmals zu Klarstellung:

Gültig: Ein Schlüssel ist gültig, wenn wir uns von der Echtheit des Schlüssels überzeugt haben.

Vertrauen: Wir vertrauen einem Schlüsselinhaber, dass er nur Schlüssel unterschreibt, bei denen er sich persönlich von der Echtheit überzeugt hat. Wir vertrauen einem Dritten, dass er die Gültigkeit der Schlüssel für uns herstellt.

4 PGP benutzen

PGP wird im wesentlichen zum effizienten Austausch verschlüsselter Dateien verwendet. Dies schließt natürlich den Austausch mit sich selber (also die Speicherung auf der eigenen Festplatte) ein. Es gibt drei Möglichkeiten, eine Datei mit PGP zu „behandeln“:

- ❒ Verschlüsseln
- ❒ Signieren
- ❒ Verschlüsseln und Signieren
- ❒ Mit dem Programm PGPTools

kann (in Abb. 4 von links nach rechts) das Schlüsselverwaltungsprogramm gestartet,

- ❒ eine Datei verschlüsselt,
- ❒ eine Datei signiert,

Abbildung 4: Die eigentliche PGP Oberfläche in Form des Programms PGPTools

- ❒ eine Datei verschlüsselt und signiert,
- ❒ eine Datei entschlüsselt/geprüft sowie
- ❒ eine Datei gelöscht und
- ❒ der freie Speicherplatz auf der Festplatte gelöscht werden.

Außerdem besteht über Kontext-Menues im Explorer bzw. über das Programm PGPTray im Systemfeld der Task-Leiste (Schloss-Symbol) Zugriff auf die PGP Funktionalität. Über das Programm PGPTray lassen sich auch alle Dateioperationen auf die Zwischenablage anwenden.

Zum Verschlüsseln benötigt der Absender den öffentlichen Schlüssel des Empfängers. Zum Signieren wird der eigene private Schlüssel benötigt. Eine signierte Datei ist im Prinzip noch im Klartext lesbar. Damit kann auch ein Empfänger, der nichts über Digitale Signaturen weiß, die Datei/E-Mail noch lesen. Er wundert sich nur über den „Datenmüll“ am Ende der Datei/E-Mail.

Charakteristisch für eine mit PGP verschlüsselte Datei sind dabei die Anfangszeile `—-BEGIN PGP MESSAGE—-` und die Endzeile `—-END PGP MESSAGE—-`. Wird dagegen ein Text mit PGP nur signiert, so sieht das Ergebnis wie in Abb. 5 gezeigt aus.

Die Datei ist nach wie vor auch ohne Wissen über PGP lesbar. Die Signatur kann nur mit PGP oder GnuPG geprüft und verifiziert werden. Zur Interpretation ist es wichtig zu wissen, dass nur der Text zwischen `—-BEGIN PGP SIGNED MESSAGE—-` und `—-BEGIN PGP SIGNATURE—-` signiert ist. Alles, was außerhalb dieser beiden Zeilen steht, ist nicht mit signiert.

Die Korrektheit geprüfter Signaturen wird in einem speziellen Fenster PGPlog *(Abb. 6)* angezeigt:

Die beiden ersten Einträge zeigen eine gültige Signatur bei der Prüfung an. Die letzte Zeile zeigt eine ungültige Signatur an. Eine ungültige Signatur kann erzeugt werden, indem an dem signierten Text eine beliebige Änderung vorgenommen wird.

5 PGP und Schlüsselserver

In die drei Feldern unter dem Servernamen tragen wir die Kriterien der Suche ein.

Ein Klick auf die Schaltfläche „Suchen“ liefert dann als Ergebnis der Suche die beiden Schlüssel für den Demo User *(Abb. 7)*.

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

Dies ist ein ASCII-Text

—-BEGIN PGP SIGNATURE—-
Version: PGP Personal Privacy 6.5.1 Int.

iQA/AwUBOIeAS6qkiqJWn1lUEQJkOACggSXLwsCghq22B+YuJpfYUMbVc
9YAmgPm
989Jv5KmE6lv1hKQzNerphGM=c5YG
—-END PGP SIGNATURE—-
```

Abbildung 5: Ein mit PGP signierter Text

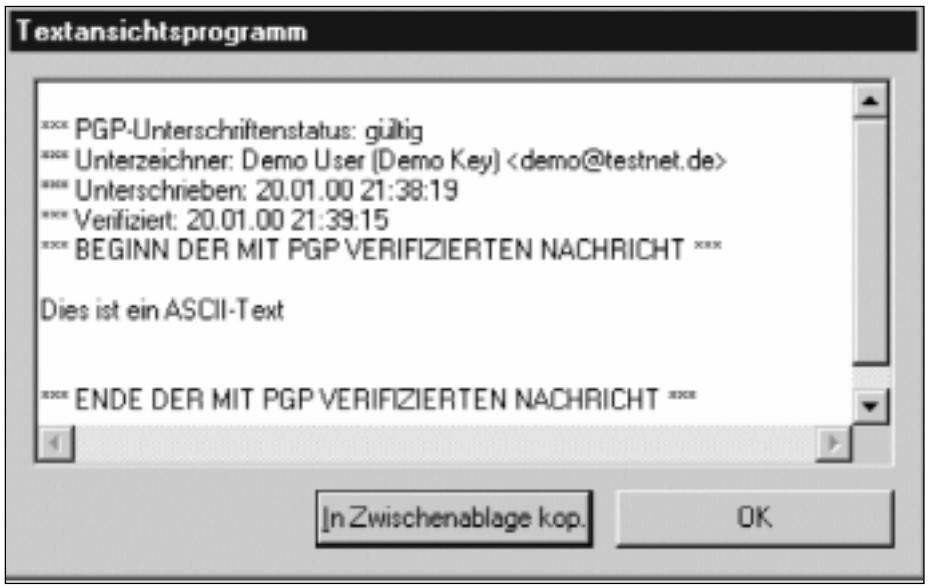

Abbildung 6: Das Ergebnis einer Signaturprüfung. Nur der Text zwischen den Zeilen mit “VERIFIZIERTER NACHRICHT” ist signiert.

Abbildung 7: Der Dialog, um auf einem Schlüsselserver nach Schlüsseln zu suchen. Im unteren Teil werden zwei gefundene Schlüssel angezeigt.

Um unseren eigenen Schlüssel auf den Server zu bekommen, markieren wir unseren Schlüssel und schicken ihn dann über den Menu-Punkt „Server->Senden an ...“ an den Schlüsselserver.

6 Schlüsselserver

Die verfügbaren Schlüsselserver stellen mit ihrer Funktionalität primär auf das Web Of Trust ab. In einer Firma werden jedoch CA-Strukturen benötigt, so dass dort der Schlüsselserver von NAI[13] bevorzugt werden sollte.

[13] eine Freeware Version (für nicht kommerzielle Anwendung) kann ohne Verletzung der US-Exportvorschriften von http://web.mit.edu/network/pgp.html heruntergeladen werden.

Dieser Server kann so konfiguriert werden, dass er zum Upload nur Schlüssel zulässt, die zertifiziert sind, das heißt, die von einem bestimmten Schlüssel unterschrieben sind. Damit geht er in seiner Unterstützung für CA Strukturen am weitesten.

Da die öffentlichen Schlüssel personenbezogene Daten im Sinne des BDSG sind, kann ein Arbeitgeber diese nicht ohne Einwilligung des Arbeitnehmers an Dritte übermitteln. Eine Veröffentlichung auf einem Schlüsselserver im Internet ist aber eine Übermittlung im datenschutzrechtlichen Sinn. Sobald auf diese öffentlichen Schlüssel auch von außerhalb der Einrichtung zugegriffen werden kann, ist der Tatbestand einer Übermittlung an Dritte erfüllt. Da es sich hier nicht um die gezielte Weitergabe an einen genau bezeichneten Dritten handelt, sondern die Daten jedem zugänglich gemacht werden, sind besonders strenge Maßstäbe anzulegen. Die schriftliche Einwilligung der Beschäftigten ist erforderlich.

Der Gesetzgeber hat im BDSG geregelt, dass die Übermittlung und Verarbeitung personenbezogener Daten erlaubt ist, wenn sie „aus allgemein zugänglichen Quellen entnommen werden können„. Ein Schlüsselserver im Internet stellt definitiv eine „allgemein zugängliche Quelle„ dar und schafft damit den Erlaubnistatbestand für eine freie Nutzung der Schlüssel. Deshalb muss hier besonders sorgfältig überlegt werden, welche Daten wie angeboten werden. Gerade auch vor dem Hintergrund einer Kommerzialisierung des Internets ist momentan nicht zu übersehen, wozu die personenbezogenen Daten aus solchen Schlüsselservern in Zukunft benutzt werden. Eines Tages werden sich auch Adresshändler im Internet intensiv „bedienen“. Ein durch öffentliche Schlüssel „aufgepepptes“ Telefon- oder Adressbuch ist gut vorstellbar.

Es ist deshalb durchaus sinnvoll, einen firmeninternen Schlüsselserver aufzubauen, auf den nur Beschäftigte zugreifen können. Der Schlüsselserver ist entweder durch eigene Konfigurationsmöglichkeiten, wie der NAI Server, oder durch eine Firewall zu schützen.

Für einen Testbetrieb ist der PKS-Server gut benutzbar, für einen Produktionsbetrieb ist der NAI Server vorzuziehen. □

PGP steht im Internet in folgenden Versionen zur Verfügung:

Für private nicht kommerzielle Zwecke kann von der Internationalen NAI PGP http://www.pgpinternational.com die Version PGP 6.5.1 deutsch heruntergeladen werden. Die aktuelle englische Freewareversion 6.5.2a findet sich auf http://www.pgpi.org. Auf dieser Webseite sind auch zahlreiche Links für Zusatztools und weiterführende Informationen.

Für geschäftliche Anwendungen findet sich auf der NAI-Webseite auch eine Evaluierungsversion von PGP Business Edition 6.5.1 deutsch mit beschränkter Laufzeit (30 Tage).