OpenPGP

Lutz Donnerhacke
Individual Network Root-CA
Lutz.Donnerhacke@Jena.Thur.De

PGP-Key: 2048 bit / KeyID 39F37F5D
A4C1 508F 00D9 2860 70BB 0B5D D93A 0BB6

19. Mai 1998

Zusammenfassung

Stand der OpenPGP-Entwicklung unter Berücksichtigung der neuen Features.

1 Das Projekt 'IN-Zertifizierungshierarchie'

1.1 Individual Network e. V.

Der Individual Network e. V. (IN) [1] betreibt seit Ende 1996 den Aufbau von Zertifizierungsstellen (IN-CAs) für öffentliche Schlüssel, vor allem für öffentliche PGP-Schlüssel. Grundlage sind die persönlichen Kontakte in den regionalen IN-Domains. Bundesweit gibt es gut 50 solcher Vereine und Gruppen mit insgesamt knapp 9 000 angeschlossenen Rechnern und über 100 000 Teilnehmern.

1.2 Struktur der Hierarchie

Die IN-Zertifizierungsinfrastruktur [2] umfaßt konzeptionell eine *Toplevel Certification Authority* oder auch kurz „Root-CA" als oberste Zertifizierungsinstanz sowie entsprechende Zertifizierungsstellen in den einzelnen IN-Domains. Regionale Zertifizierungsstellen gibt es inzwischen in Berlin, Thüringen, Mittelhessen, Brandenburg, Südwest-Sachsen, Kiel, München, Karlsruhe und Franken (Stand Januar 1998). Darüberhinaus befinden sich entsprechende Stellen in fünf weiteren IN-Domains im Aufbau. Zusätzlich betreibt die IN-Root-CA noch eine „überregionale" IN-CA für die Zertifizierung der Gremien des Individual Network (Geschäftsstelle, Vorstand, Beirat usw.). Sie dient zugleich auch als Anlaufstelle für alle Teilnehmer aus solchen Gegenden, in denen es noch keine eigene regionale IN-CA gibt. (Eine detaillierte Darstellung der IN-Zertifizierungsinfrastruktur findet sich in [3].)

1.3 PGP-Einsatz

Im Rahmen der Arbeiten in den IN-Zertifizierungsstellen, die sich zu Beginn auf PGP-Schlüssel konzentrierte, wurde zunächst die zum damaligen Zeitpunkt aktuelle PGP-Version PGP2.6.3ia eingesetzt [6]. Dabei traten jedoch einige Schwierigkeiten mit dieser Version zutage, die der Anlaß für eine Weiterentwicklung der Software waren, die zu einer eigenen PGP-Variante [7] führten. Diese implementiert einige Funktionen im Bereich der Schlüsselverwaltung, die schon in der Dokumentation zum Original-PGP beschrieben, aber nie realisiert worden waren.

Wesentliche Teile der Funktionalität dieser IN-PGP-Version wurden dann etwas später in den Entwurf eines 'OpenPGP'-Standards [15] übernommen, der darüberhinaus auch einige konzeptionelle Schwächen der existierenden PGP-Versionen vermied.

2 OpenPGP

2.1 Ausgangslage

Mit der Einführung von PGP5 wurde ein scharfer Bruch bzgl. der Interoperabilität von PGP vorgenommen. Ursachen dieses Bruchs sind:

- Angreifbarkeit der KeyID [17] (vgl. Abb. **??**)
- Angreifbarkeit des Fingerprints [17]
- Entdeckte Schwäche in MD5 (Hans Dobbertin, BSI [10])
- Patentstreitigkeiten

Die Idee mit OpenPGP entstand bei der Vermarktung von PGP5. Die Wirtschaft forderte standardkonforme Software, die Konkurrenz von Netscape war auf dem Weg zu einem Standard für ihr Produkt S/MIME [14], also wollte PGP Inc. nachträglich einen Standard schaffen. „Un“glücklicherweise wählte man den Weg eines Internet *Request for Comments* (RfC). Die entsprechenden Arbeitskreise der IETF stehen jederman offen, so daß eine weltweite Mitarbeit möglich war.

Da PGP Inc. die Standardisierungsbemühungen schleifen ließ und zudem die „Aufweichung“ der kommerziellen Version (Stichwort ‘Message Recovery’, also die Mitlesemöglichkeit für Dritte) forderte, formierte sich in England, Neuseeland und Deutschland eine kleine Gruppe außerhalb von PGP Inc., die die Entwicklung des Standards selbst in die Hand nahm. PGP5 wird den OpenPGP-Standard *nicht erfüllen* können.

Der aktuelle Stand ist, daß die IN-Root-CA Co-Autor des informalen Drafts ist [20] und eine formale Spezifikation sowie die Referenzimplementation vollständig auf dem Tisch hat [16]. Die Möglichkeiten sind also nahezu unbegrenzt, zumal PGP Inc. nach dem Aufkauf durch Network Associates Inc. [13] nicht besonders handlungsfähig ist. Diese Referenzimplementation unter *GNU Public License* soll BSI-zertifizierbar sein, so daß sie den Anforderungen des Signaturgesetzes [21] und der Signaturverordnung [23] genügt.

Über die normalen und erwarteten Fähigkeiten hinaus versucht die OpenPGP-Implementation auch gerichtsfeste Zeitstempeldienste [18] und darauf basierend ein anonymes, übertragbares, bankenfreies, nachbezahltes Geldprotokoll [19] zu implementieren. Inwieweit Unterstützung elektronischer Wahlen eingebaut wird, ist eine Frage der Erfordernisse.

2.2 Neue Features

OpenPGP definiert sich als einheitliche Spezifikation für alle PGP-Versionen. Es wird weitestgehende Abwärtskompatibilität gefordert. Sie behandelt im Prinzip nur das Nachrichtenformat. Es werden neue Algorithmen zugelassen, z.B. ElGamal, DSA als asymmetrische Verfahren; DES3, CAST5, BLOWFISH, SAFER als symmetrische Verfahren; SHA1, HAVAL, RIPEMD160, ... als Hash-Verfahren. (Quellen zu den meisten dieser Verfahren in [10])

2.2.1 Paketstruktur

Alle Pakete haben eine flexible, aber stets definierte Länge, im Gegensatz zum alten Verfahren, bei dem auch eine semantikabhängige Länge auftauchen konnte. Damit und mit einigen Zusatzpaketen wird OpenPGP one-pass-fähig und kann so auf die sicherheitsgefährdenden temporären Dateien verzichten, mit denen es jetzt noch arbeiten muß.

Im Rahmen der Umstrukturierung wurde Platz für mehr Pakettypen geschaffen, und die Zertifikate wurden um Erweiterungen ergänzt. Es gibt neue Algorithmen, die KeyID und den Fingerprint zu bestimmen, die den bekannten Angriffen standhalten.

Das Paradigma, daß KeyIDs und Fingerprints eindeutig zu sein hätten, wurde verworfen. Diese sind nunmehr nur noch eine Maske auf alle möglichen

IDs. Eine KeyID `0x0...0` bedeutet also: *„Probiere alle möglichen Schlüssel durch!“* Dies gestattet voll anonyme Mailzustellung, da die Empfänger-KeyID nicht mehr als Teil der verschlüsselten Nachricht verschickt werden muß, wie das bislang der Fall war.

Eine UserID muß selbstunterschrieben sein, sonst ist sie ungültig. Dies verhindert Unschönheiten wie von fremde Personen an die UserID angefügte Beleidigungen. Eine zurückgezogene Selbstsignatur widerruft die UserID. Zusätzlich können Schlüssel auch ohne UserID signiert werden, beispielweise um Schlüsselparameter zu setzen.

2.2.2 Neue Pakete

Das neue Paket **‘Symmetrische Verschlüsselung’** bietet einen verbesserten Mantraschutz und endlich wieder die klassische Verschlüsselung. Es ist nunmehr möglich, daß ein und dasselbe Mantra andere Sessionkeys verschlüsselt. Dies bedeutet speziell, daß das gleiche Mantra verschiedene private Schlüssel verschieden verschlüsselt. Wörterbuchangriffen wird ebenfalls vorgebeugt.

Das Paket **‘One-Pass Signatur’** gestattet es, ohne temporäre Dateien Signaturen zu prüfen, da vorab die benötigten Algorithmen bekannt sind. Ebenso wird eine Schachtelung möglich. Man kann also mit OpenPGP ein unterschriebenes Dokument unterschreiben. Dies bietet neue Möglichkeiten der Beglaubigung gemäß traditionellem Rechtsverkehr.

Mit **Subschlüsseln** wird einem Hauptschlüssel ein beliebiger Satz von Unterschlüsseln zugeordnet. Diese Schlüssel werden als Einheit betrachtet. So kann man sehr häufig (täglich!) den Verschlüsselungsschlüssel wechseln. Hiermit kann man verbesserte ‘Forward Security’, im Extremfall sogar ‘Perfect Forward Security’ erzielen. Für den Zertifizierungsbetrieb ist dies interessant, weil sich so *alle notwendigen Zwischenschlüssel* (Unterschriften von Textdokumenten, monatlicher Verschlüsselungsschlüssel, ...) *unter einem Zertifizierungsschlüssel subsumieren lassen.*

Ein spezielles **Markerpaket** kann sogar bei Binärkompatibilität mit ASN.1 in zukünftigen Versionen von externen Scripts, wie z. B. Procmail, immer noch als PGP-Paket erkannt werden.

2.2.3 Signaturerweiterungen

Eine Signatur kann deutlich mehr ausdrücken. Dabei wird unterschieden, ob die Erweiterung mit unterschrieben wurde oder nicht. Mit unterschriebene Erweiterungen können als ‘kritisch’ markiert sein. Software, die eine 'kritische' Erweiterung nicht verarbeiten kann, muß die Signatur verwerfen. Unkritische unbekannte Erweiterungen dürfen hingegen ignoriert werden.

Neu sind die Möglichkeiten, einen *Gültigkeitszeitraum für Unterschriften, Zertifikate und Schlüssel* festzulegen. Darüberhinaus kann man einen Schlüssel als *exportierbar* (aus einer Umgebung) und *rückziehbar* markieren oder eben dieses verbieten.

Es ist möglich, die Wirkung auf den Keyservern zu limitieren. So kann der Nutzer einen Standardkeyserver definieren. Dieser ist dann bevorzugter Anlaufpunkt für Onlineprüfungen hinsichtlich eventueller Widerrufe. Ebenso kann er Fremd-Updates des eigenen Schlüssels auf den Keyservern verbieten.

Implementiert eine Software nicht alle möglichen Algorithmen, sondern nur eine Auswahl, so kann sie die implementierten angeben. Das ist jedoch mit Vorsicht zu genießen, da hier auch Hash- und Signaturalgorithmen angegeben sein können, die u.U. brechbar sind (z.B. ROTn). *Dies gefährdet eventuell Dritte!*

Durch weitere Zusatzinformationen wie Fingerprint, KeyID, UserID, Finger, URL kann man nähere Informationen zum verwendeten (Unterschriften-)Schlüssel bereitstellen.

Der *Verwendungszweck* von Schlüsseln kann eingeschränkt werden auf:

- Zertifizierungen (andere Schlüssel/UserID Paare unterschreiben)
- Signieren (Texte/Binaries unterschreiben)
- Verschlüsselung für Kommunikationsanwendungen
- Verschlüsselung für lokale Speicherungen
- Gruppenschlüssel (für Verschlüsselungen an eine Gruppe von Personen)

Einige Kombinationen von Verwendungszwecken sind unzulässig. So kann beispielsweise ein Gruppenschlüssel nicht unterschreiben. Gruppenschlüssel für lokale Speicherung sind für Firmen sinnvoll, die Zugriff auf die lokal gespeicherten Daten benötigen, ohne die komplette Kommunikation mitlesen zu wollen. Dies macht das Message Recovery von PGP Inc. unnötig und beseitigt ein so erhebliches „Sicherheitsloch“.

Das Message Recovery ist nicht verschwunden. Es wird aktiv bekämpft, d.h. es ist gefordert, dem Nutzer eine Fälschung der Message Recovery Keys anzubieten. Diese Fälschungen sind maschinell nicht erkennbar, selbst wenn versucht wird zu entpacken.

Für den Zertifizierungsbetrieb interessant sind die Möglichkeiten der *Hierarchieunterstützung.* Man kann eine *regular expression* für diese Hierarchie und eine Stufe der Hierarchie festlegen. Damit ist es möglich alle benötigten Hierarchien ins Web of Trust einzubetten, ohne zwangsweise auf unnötige Hierarchisierung zurückgreifen zu müssen. Crosszertifizierungen sind dadurch ebenfalls problemlos möglich.

Für viele Anwendungen wird sich trotz allem kein fertiges Paket finden. Da jeder Unterschreibende frei in der Gestaltung von Zusatzpaketen ist – derartige *Freiformen* sind im Standard vorgesehen –, können auch ausgefallene Varianten des normalen Rechtsverkehrs bis ins Detail abgebildet werden. Eine Registrierung von *Object Identifiern* ist dabei *nicht* notwendig.

2.3 Politik

CAs ($\neq$ „Trusted Third Parties" [TTPs] = Schlüsselhinterlegungsstellen (EU-Sprachgebrauch)) dürfen u. U. nur nach Genehmigung arbeiten. Gewisse Hürden hinsichtlich Personal und Verfahrenvorschriften sind politisch erwünscht, um die Zahl der Zertifizierungsstellen klein zu halten und die Zertifizierungsinfrastruktur bei Bedarf leichter in eine *Key-Escrow*-Struktur umwandeln zu können. Chipkarten (TeleSec) erfüllen bislang als einzige Produkte die technischen Anforderungen, implementieren allerdings oft Schlüsselhinterlegungsprotokolle.

Insofern könnte OpenPGP durch die „Hintertür" der CA-Zulassung eine Kryptoregulierung systemkonform unterlaufen.

3 IN-CA Schlüssel

2048-bit-Schlüssel mit KeyID 0x19980101, erstellt am 1998/01/12 mit dem Fingerabdruck

```
B3 06 9A 8D 38 04 3C 75  41 32 EE DC 8B 7D 61 0D
```

in-ca@individual.net SIGN EXPIRE:1999-12-31 Root CA des Individual Network e.V.

Literatur

[1] http://www.individual.net/

[2] https://www.in-ca.individual.net/

[3] I. Camphausen: „Erfahrungen beim Aufbau einer bundesweiten Zertifizierungsinfrastruktur für den Individual Network e. V.". Erscheint im Tagungsband *Sicherheit in Informationssystemen SIS'98*

[4] S. Schumacher, PGP 2.6.3(i) „international“ (Sourcecode), 18. Januar 1996. ftp://ftp.cert.dfn.de/pub/tools/crypt/pgp/unix/pgp263is.tar.gz

[5] S. Schumacher, PGP 2.6.3i patches, 4. März 1996, news:alt.security.pgp, Msg-ID <4hekp4$k5l@surt.ifi.uio.no>

[6] Datei pgformat.doc" der PGP2.6.3(i)a-Distribution ([5])

[7] ftp://ftp.individual.net/pub/doc/IN/IN-CA/pgp/pgp236in/

[8] S.Kelm, „Zertifizierungsrichtlinien des Projekts ‘PCA im DFN”“, in: DFN-PCA (Hg.), *DFN-Bericht Nr.82*, Hamburg, 29. April 1997. http://www.pca.dfn.de/dfnpca/policy/lowlevel.html

[9] *Pretty Good Privacy 5.0 – Platform Independent Source Code,* PGP Inc., 2nd ed., 9. Juli 1997. ftp://ftp.ifi.uio.no/pub/pgp/5.0/international/unix/pgp50i-unix-src-b8a.tar.gz

[10] H. Dobbertin, „The Status of MD5 After a Recent Attack“, in: RSA Labs, *CryptoBytes,* Vol. 2, No. 2, Summer 1996, 1–6

[11] *Digital Signature Standard DSS,* Federal Information Processing Standards Publication (FIPS PUB) 186, United States National Institute of Standards and Technology, 19. Mai 1994. http://csrc.nist.gov/fips/fips186.pdf

[12] IEEE-Mailingliste zur Erarbeitung des Standards P1363: ‘stds-p1363’ / ‘stds-p1363-discuss’ on majordomo@mail.ieee.org, http://stdsbbs.ieee.org/groups/1363/

[13] „Network Associates Acquires Pretty Good Privacy (PGP),“ Press Release, Network Associates, 1. Dezember 1997, *http://www.pgp.com/newsroom/prel47.cgi*

[14] B.Ramsdell, *S/MIME Version 3 Message Specification,* Internet-Draft, 4. November 1997. http://www.imc.org/draft-ramsdell-smime-msg

[15] http://www.imc.org/ietf-open-pgp/

[16] ftp://ftp.individual.net/pub/doc/IN/IN-CA/pgp/OpenPGP/

[17] comp.security.pgp FAQ, Frage 4.10

[18] https://www.iks.jena.de/mitarb/lutz/logfile/

[19] `https://www.iks.jena.de/mitarb/lutz/ecash/`

[20] J. Callas, L. Donnerhacke, H. Finney, R. Thayer, *OPFormats – OpenPGP Message Format,* Internet-Draft, November 1997. `ftp://ftp.nordu.net/internet-drafts/draft-ietf-openpgp-formats00.txt`

[21] *Gesetz zur digitalen Signatur (Signaturgesetz – SigG)*, in [22], `http://www.iid.de/rahmen/iukdgbt.html#a3`

[22] *Gesetz zur Regelung der Rahmenbedingungen fur Informations- und Kommunikationsdienste (Informations- und Kommunikationsdienste-Gesetz – IuKDG)* vom 22.07.1997, BGBl I 1997 Nr. 52, S. 1870–1880. `http://www.iid.de/rahmen/iukdgbt.html`

[23] *Verordnung zur digitalen Signatur (Signaturverordnung – SigV)*, Beschluß der Bundesregierung der Bundesrepublik Deutschland, 8. Oktober 1997. `http://www.iid.de/rahmen/sigv.html`