

Konzepte von Betriebssystem-Komponenten: PGP

Dreweke Alexander
a.dreweke@gmx.de

10.06.2002

1 Allgemeines

Bei PGP handelt es sich um ein sogenanntes Hybrid Verschlüsselungssystem. Es vereint die Vorzüge der symmetrischen und asymmetrischen Verschlüsselung in sich.

Durch die Verwendung der symmetrischen Verschlüsselung für die eigentliche Nachricht, hat man auch bei geringerer Schlüssellänge (128 bis 256 Bit) eine sehr hohe Sicherheit die gleichzeitig, im Vergleich zur asymmetrischen Verschlüsselung (1024 bis 4096 Bit Schlüssellänge), sehr schnell von statten geht.

Das Schlüsselaustauschproblem wird durch die Verwendung von asymmetrischen Verschlüsselungsverfahren gelöst. Hierbei werden bei der Schlüsselgenerierung ein Private- und ein Public-Key erstellt. Mithilfe des Public-Keys, der allen potentiellen Kommunikationspartnern frei zugänglich ist, werden Nachrichten verschlüsselt, die nur mit dem Private-Key wieder entschlüsselt werden können.

1.1 verwendete Algorithmen in PGP

symmetrische Algorithmen: 3DES, AES, CAST, IDEA, TWOFISH

asymmetrische Algorithmen: RSA (bis 3072 Bit Schlüssellänge), DH/DSS (bis 4096 Bit Schlüssellänge)

Hash-Algorithmen: SHA1

1.2 verwendete Algorithmen in GPG

Bei GPG handelt es sich um eine frei Implementierung von PGP, GPG verhält sich genau (Abweichungen sind in Text beschrieben) wie PGP, verwendet aber teilweise andere Algorithmen.

symmetrische Algorithmen: 3DES, CAST5, BLOWFISH, RIJNDAEL, TWOFISH

asymmetrische Algorithmen: RSA, RSA-E, RSA-S, ELG-E, DSA, ELG

Hash-Algorithmen: MD5, SHA1, RIPEMD160

2 Ver- & Entschlüsselung einer Nachricht

Eine Nachricht soll so verschlüsselt werden, dass sie nicht von unbefugten Dritten gelesen werden kann.

2.1 Verschlüsseln von Nachrichten

Eine Nachricht (Klartext) wird von PGP zuerst dahingehend überprüft, ob sie komprimiert werden kann (nicht komprimiert werden bereits komprimierte Daten oder Daten die dafür zu kurz sind), um auf diese Art und Weise Speicherplatz und Zeit zu sparen. Ein weiterer Vorteil der Komprimierung besteht darin, dass Redundanzen aus der Nachricht entfernt werden und so der Schutz vor kryptoanalytischen Angriffen deutlich vergrößert wird (der zusätzliche Zeitaufwand für das Komprimieren ist somit durchaus gerechtfertigt).

Anschließend wird der Klartext bzw. der komprimierte Klartext mit Hilfe eines symmetrischen Verschlüsselungsverfahrens verschlüsselt. Der so entstehende verschlüsselte Klartext heißt Chiffre. Der für die symmetrische Verschlüsselung notwendige Schlüssel (Session-Key) wird durch einen Zufallszahlengenerator erstellt. Dieser Session-Key wird nun anschließend mit dem Public-Key des Empfängers verschlüsselt.

Soll die Nachricht an mehrere Personen geschickt werden wird nur der Session-Key für jeden Empfänger mit dessen Public-Key einzeln verschlüsselt. Die Nachricht an sich muss nicht noch einmal verschlüsselt werden dies spart sowohl Zeit als auch Speicherplatz. Bevor die PGP-Nachricht (Chiffre und verschlüsselte(r) Session-Key(s)) via Internet übertragen wird sollte sie noch in die sogenannte ASCII-Rüstung (siehe 2.3) gesteckt werden, um eine unverfälschte Übertragung zu gewährleisten.

2.2 Entschlüsseln von Nachrichten

Eine verschlüsselte Nachricht wird zuerst aus der ASCII-Rüstung ausgepackt (sofern vorhanden). Anschließend wird die Nachricht mit dem Session-Key entschlüsselt. Diesen erhält man indem man mit dem eigenen Private-Key den mit dem Public-Key verschlüsselten Session-Key entschlüsselt. Wurde der Klartext von PGP komprimiert so wird dieser noch entpackt.

2.3 ASCII-Rüstung

PGP bedient sich bei der Darstellung des Chiffretextes, aller Zeichen die mit einem 8-Bit Wert gespeichert werden können. Bei der Übertragung des Chiffretextes über das Internet besteht die Gefahr, dass die Nachricht über einen Server geleitet wird, der missachtet, dass es 8-Bit Zeichensätze gibt, d.h. für ihn ist das erste Bit eines Bytes immer 0. Wird die Nachricht über einen derartigen Server geleitet, wird sie derart verfälscht, dass eine Entschlüsselung unmöglich ist.

Aus diesem Grund bedient man sich der sogenannten ASCII-Rüstung. Nach der Verschlüsselung werden die Daten maskiert: Aus drei Byte=24 Bit werden vier Byte mit je zwei führenden 0-Bits und sechs Daten-Bits. Eine derart maskierte Nachricht kann bei der Übertragung nun nicht mehr verfälscht werden.

3 Signieren & Verifizieren einer Nachricht

Bei einem Nachrichtenaustausch muss zum einen sichergestellt werden, dass die Nachricht nicht unbemerkt verändert wurde, zum anderen muss die Identität des Absenders eindeutig

bestimmbar sein.

3.1 Signieren von Nachrichten

Von einer Nachricht wird zunächst ein Hash-Wert gebildet. Dieser wird anschließend mit dem eigenen Private-Key verschlüsselt (diesen so verschlüsselten Hash-Wert nennt man auch Signatur). Dadurch kann der verschlüsselte Hash-Wert wieder durch die Entschlüsselung der Signatur mit dem Public-Key gewonnen werden. Bei asymmetrischen Verschlüsselung (wie z.B: RSA) können Nachrichten die mit dem Privat-Key verschlüsselt wurden mit dem entsprechenden Public-Key wiederhergestellt werden. Dabei kann je nach Verfahren der gleiche Algorithmus zum ver- und entschlüsseln (wie bei RSA) zum Einsatz kommen oder wie bei DSA unterschiedliche Algorithmen.

3.2 Verifizieren von Nachrichten

Eine signierte Nachricht wird verifiziert indem man die Signatur mit dem Public-Key des Absenders entschlüsselt und diesen entschlüsselten Hash-Wert mit dem Hash-Wert vergleicht den man selbst erstellt. Es müssen jeweils dieselben Hash-Funktionen verwendet werden. Stimmen die beiden Hash-Wert überein, so ist die Nachricht unverändert. Dadurch, dass der mitgeschickte Hash-Wert mit dem Public-Key des Absenders entschlüsselt werden konnte ist sichergestellt, dass die Nachricht auch wirklich vom richtigen Absender stammt.

3.3 Hash-Funktion

Bei Hash-Funktionen handelt es sich um Einwegfunktionen, die von einem großen Wertebereich auf einen kleineren abbildet. An Hash-Funktionen für digitale Unterschriften werden folgende Forderungen gestellt:

1. Die Ausgabe muss eine feste Länge (z.B: 160 Bit) haben.
2. Auch bei geringfügiger Änderung der Eingangsinformation muss ein völlig veränderter Ausgabewert erzeugt werden.
3. Mithilfe des Hash-Wertes dürfen keinerlei Rückschlüsse auf die eigentliche Nachricht möglich sein.
4. Es darf keine zwei sinnvollen Nachrichten geben, die den selben Hash-Wert erzeugen.

3.4 Key-Server

Um den Aufwand beim Austausch der Public-Keys zu minimieren wurden im Internet zentrale Anlaufstellen geschaffen. Diese so genannten Key-Server ermöglichen es jedem seinen Public-Key für jedermann zugänglich zu machen. Benötigt man nun den Public-Key einer Person, so kann man diesen direkt von einem Key-Server erhalten. Hierzu sendet man den Hash-Wert des gewünschten Public-Key, den sogenannten Fingerprint, an den Server und erhält daraufhin den passenden Schlüssel. Diesen Fingerprint findet man häufig auf Homepages, Visitenkarten oder auch in eMail-Absendern.

4 Zertifikate & Web-of-Trust

Zertifikate sollen sicherstellen, dass ein Public-Key auch wirklich zu der realen Person gehört, die vorgibt den Schlüssel erstellt zu haben. Ein Zertifikat besteht aus Daten, die dem öffentlichen Schlüssel einer Person hinzugefügt werden. Anhand dieser Daten kann festgestellt werden, ob ein Schlüssel gültig ist. Damit können dann Versuche, den Schlüssel einer Person durch den einer anderen auszutauschen (man-in-the-middle-attack), vereitelt werden. In PGP kommen das X.509- und PGP-Zertifikatsformat zum Einsatz.

4.1 X.509-Zertifikatsformat

Eines der bekanntesten Zertifikatsformate ist das X.509. Es ist nach dem internationalen Standard ITU-T X.509 genormt und kann somit theoretisch für verschiedene Anwendungen verwendet werden. Für die Ausstellung eines Zertifikates muss überprüft werden, ob der öffentliche Schlüssel und der Name des Schlüsseleigentümers zusammengehören. Dies geschieht durch eine zentralen Zertifizierungsinstanz oder durch eine Person, die von einer Certificate Authority (CA) benannt wurde.

Ein X.509-Zertifikat ist eine Sammlung von Standardfeldern, die Informationen über einen Benutzer und die entsprechenden öffentlichen Schlüssel enthalten. Der X.509-Standard legt fest, welche Informationen in das Zertifikat aufgenommen werden und beschreibt, wie es entschlüsselt wird (das Datenformat). X.509-Zertifikate enthalten die folgenden Daten:

1. X.509-Versionsnummer
2. öffentlicher Schlüssel des Zertifikatsinhabers
3. Seriennummer des Zertifikats (ausstellerspezifisch)
4. eindeutige Kennung des Zertifikatsinhabers (z.B: Name)
5. Gültigkeitsdauer des Zertifikats
6. eindeutiger Name des Zertifikatsausstellers
7. digitale Unterschrift des Ausstellers
8. Kennung des Unterschriftenalgorithmus

4.2 PGP-Zertifikatsformat

Bei einem PGP-Zertifikat kann jeder die Rolle der überprüfenden Person übernehmen. Außerdem können, im Gegensatz zum X.509-Zertifikat, auch mehrere Unterschriften enthalten sein. Dies ist aus mehreren Gründen sinnvoll: zum einen ist die Wahrscheinlichkeit, dass mehrere Leute bei der Überprüfung der Identität des Schlüsseleigentümers ein Fehler unterläuft, wesentlich geringer als wenn ein Schlüssel nur von einer zentralen Stelle geprüft wird; zum anderen kann man als Benutzer selbst entscheiden, in wie weit man den einzelnen Personen, die den Schlüssel zertifiziert haben, vertraut. Ein PGP-Zertifikat enthält mindestens die folgenden Informationen:

1. PGP-Versionsnummer
2. öffentlicher Schlüssel des Zertifikatsinhabers
3. Daten des Zertifikatsinhabers

4. digitale Unterschriften der verifizierenden Personen
5. Gültigkeitsdauer des Zertifikats
6. bevorzugten symmetrischen Verschlüsselungsalgorithmus für die Schlüssel

4.3 Web-of-Trust

Web-of-Trust stellt eine Kombination des direkten und des hierarchischen Vertrauens dar. Direkt bedeutet, dass der Benutzer auf die Gültigkeit eines Schlüssels vertraut, da dessen Herkunft bekannt ist, hierarchisch, dass er einer Anzahl von Root-Zertifikaten vertraut und somit auch allen Schlüsseln, deren Vertrauen auf einem direkten Weg abgeleitet werden kann. Es handelt sich hierbei um ein kumulatives Vertrauensmodell. In einer PGP-Umgebung kann jeder Benutzer als Zertifizierungsinstantz agieren. Jeder PGP-Benutzer kann also die Gültigkeit des Zertifikats für den öffentlichen Schlüssel eines anderen PGP-Benutzers bestätigen. Ein derartiges Zertifikat ist aber nur dann für den anderen Benutzer echt, wenn die überprüfende Person von der abhängigen Partei als ein autorisierter Schlüsselverwalter anerkannt wird. Auf jedem öffentlichen Schlüsselbund eines Benutzers werden folgende Informationen gespeichert:

1. Ob der Benutzer einen bestimmten Schlüssel als gültig betrachtet
2. Die Vertrauensstufe, die der Benutzer dem Schlüssel zuordnet, d. h. die Eignung des Schlüsseleigentümers als Zertifizierungsinstantz für andere Schlüssel

In PGP werden 3 Vertrauensstufen unterschieden:

1. volles Vertrauen
2. eingeschränktes Vertrauen
3. kein Vertrauen (oder Nicht vertrauenswürdig)

Schlüssel denen noch nicht explizit eine Vertrauensstufe zugewiesen wurde, werden als nicht vertrauenswürdig eingestuft. Allerdings werden sie als Unbekannt gekennzeichnet um sie von denen mit explizitem Mißtrauen zu unterscheiden.

Darüber hinaus gibt es drei Gültigkeitsstufen:

1. gültig
2. zweitrangig gültig
3. ungültig

Die höchste Vertrauensstufe eines Schlüssels, implizites Vertrauen, ist das Vertrauen in das eigene Schlüsselpaar. Bei PGP wird davon ausgegangen, daß bei Besitz eines privaten Schlüssels auch den Aktionen der zugeordneten öffentlichen Schlüssel vertraut werden muß. Alle Schlüssel, die vom Schlüssel mit dem impliziten Vertrauen unterschrieben wurden, sind gültig (direktes Vertrauen). Um die Gültigkeit eines Schlüssels zu bestätigen, sind bei PGP eine Unterschrift der vollen Vertrauensstufe oder zwei Unterschriften der eingeschränkten Vertrauensstufe erforderlich (hierarchisches Vertrauen). GPG geht hier etwas weiter und fordert mindestens drei Unterschriften der eingeschränkten Vertrauensstufe um eine Schlüssel als gültig zu akzeptieren.

5 Protokollschwäche

Ein Problem bei den hybrid Verschlüsselungsverfahren, zu denen auch PGP gehört, ist, dass ein einfaches Sign & Encrypt bzw. Encrypt & Sign keine wirkliche Sicherheit hinsichtlich des wahren Absenders oder der Unverfälschtheit der Nachricht bietet.

5.1 Sign & Encrypt Schwäche

B hat einen Chiffretext (enthält sowohl den Klartext als auch digitale Unterschrift) von A erhalten. Wird der Klartext und A's digitale Unterschrift nun mit einem anderen Public-Key (C's) verschlüsselt und an diesen gesandt (mit gefälschtem Absender), so kann C nicht erkennen, dass die Nachricht nicht auf direktem Weg von A zu ihm gelangt ist. Der Klartext ist unverändert und somit ist A's digitale Unterschrift immer noch gültig.

5.2 Encrypt & Sign Schwäche

A sendet einen Chiffretext mit der dazugehörigen digitalen Signatur an C. Diese Nachricht wird von B abgefangen. B muss im vorhinein Kenntnis über den Inhalt der Nachricht haben, da er die Nachricht nicht entschlüsseln kann. B entfernt die Signatur von A und ersetzt sie durch die eigene, bevor er die Nachricht an C weiterleitet. Für C sieht es so aus als wäre die Nachricht direkt von B gekommen, da der Chiffretext unverändert geblieben ist.

5.3 Lösungsansätze

Für die in den beiden vorangegangenen Absätzen beschriebenen Probleme gibt es zwei unterschiedliche Ansätze, um sie zu beheben. Beide Ansätze lösen das Problem nicht wirklich, bieten aber zumindest einen Weg, der solche Fälschungsversuche sichtbar macht:

1. Eine sehr einfache Methode besteht darin, in der Nachricht selbst festzuhalten von wem sie stammt und an wen sie gehen soll. Damit ist dokumentiert wer die Nachricht für wen verfaßt hat und der Empfänger sollte damit in der Lage sein zu merken wenn, die Signatur nicht mit der des angegebenen Absenders übereinstimmt, bzw. dass die Nachricht ursprünglich nicht für ihn bestimmt war.
2. Wenn es sich bei der zu verschickenden Nachricht allerdings nicht um einen Text, sondern zum Beispiel um ein Bild bzw. um einen Text, der nicht verändert werden kann, handelt, helfen folgende Ansätze:
 - (a) Statt eines einfachen Sign & Encrypt verwendet man ein Sign & Encrypt & Sign. Dadurch wird verhindert, dass die Nachricht einfach mit einem anderen öffentlichen Schlüssel verschlüsselt wird, da dann entweder die innere Signatur nicht mehr zur äußeren Signatur paßt, bzw. die äußere Signatur als falsch erkannt wird.
 - (b) Ein anderer Ansatz funktioniert ähnlich: Statt Encrypt & Sign verwendet man Encrypt & Sign & Encrypt. Durch das zweite Verschlüsseln ist es unmöglich, die Signatur zu entfernen und durch eine andere zu ersetzen.

Dabei müssen sich aber beide Kommunikationspartner auf die Reihenfolge Sign & Encrypt & Sign bzw. Encrypt & Sign & Encrypt geeinigt haben.

Literatur

- [1] J. Callas, L. Donnerhackle, H. Finney, R. Thayer, 'OpenPGP Message Format', RFC 2440, November 1998
- [2] M. Elkins, 'MIME Security with Pretty Good Privacy (PGP)', RFC 2015, Oktober 1996
- [3] D. Atkins, W. Stallings, P. Zimmermann, 'PGP Message Exchange Formats', RFC 1991, August 1996
- [4] D. Davis, 'Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP, and XML'
- [5] S. Gerfinkel, 'PGP', Bonn, O'Reilly & Associates, Januar 1995
- [6] MAN-, INFO-Pages zu PGP und GPG