Unix I – Einführung in das Unix Betriebssystem

Stephan Thesing

Netzwerke

- PGP

PGP – Pretty Good Privacy

In vielen Bereichen ist es sinnvoll, seine Daten zu *verschlüsseln*.

Gründe:

- Andere Leute (auch Administratoren!) sollen bestimmte Dateien nicht lesen
- Der Übertragungsweg (Email) ist nicht abhörsicher
- Man will eine Möglichkeit habe, Nachrichten zu *authentifizieren*

Mit dem PGP (Pretty Good Privacy) Programm gibt es eine Möglichkeit, Daten relativ sicher zu verschlüsseln.

PGP ist (in einigen Versionen:-) frei verfügbar und weit verbreitet.

Man kann damit:

- Public-Keys erzeugen
- Nachrichten ver- und entschlüsseln
- Nachrichten signieren
- Öffentliche Schlüssel anderer Leute verwalten
- Schlüssel signieren (Web of Trust)

PGP gibt es in mehreren Versionen.

Die bei uns verbreiteten Versionen:

- PGP V2.6.3 (NetBSD CIP Rechner)
- PGP V5 (FreeBSD CIP Rechner)

Sind (teilweise) inkompatibel/unterscheiden sich in der Bedienung.

Basisfunktionen gleich.

Hier werden beide besprochen.

PGP verwendet *Public Key* Verfahren, um Daten zu ver- und entschlüsseln.

Dazu muss ein Schlüsselpaar aus *privatem* und *öffentlichem* Schlüssel erzeugt werden.

Der öffentliche Schlüssel kann (und muss) an alle verteilt werden, die einem Nachrichten schicken wollen.

Der private Schlüssel muss geheimgehalten werden und dient zum Entschlüsseln der empfangenen Nachrichten.

Die beiden Schlüssel hängen natürlich zusammen :-)

Im Prinzip kann man mit dem eingesetztem Cryptographieverfahren (RSA) nun schon Nachrichten an einen Empfänger verschlüsseln.

I.A. will man Nachrichten aber an mehrere Empfänger schicken.

Um dieselbe Nachricht nicht n-mal wegzuschicken (jeweils verschlüsselt mit den public Key des jeweiligen Empfängers) macht man einen Umweg.

Die Nachricht wird mit einem secret-key Verfahren verschlüsselt.

An die Nachricht angehängt wird nun nur dieser secret Key, n-mal verschlüsselt mit den public Keys der n Empfänger.

Jeder der Empfänger kann nun den secret Key entschlüsseln und mit diesem die Nachricht.

Neben dem Verschlüsseln einer Nachricht möchte man manchmal auch einfach deren Absender verifizieren.

PGP bietet auch dazu eine Möglichkeit.

Hierzu wird eine sog. Hashfunktion auf die Nachricht angewandt.

Diese liefert etwa eine 256 Bit Zahl (den Hash).

Diese Funktion ist nur sehr schwer wieder umkehrbar (Hash → Nachricht ist schwer)

Der Hash wird dann mit dem *privatem* Schlüssel des Absenders verschlüsselt und an die Nachricht angehängt.

Jeder, der den öffentlichen Schlüssel kennt kann nun den ursprünglichen Hash wieder erhalten (durch Entschlüsseln).

Vergleich des Hashs mit dem tatsächlichem Hash der Nachricht gibt nun Auskunft über den Signierer der Nachricht.

Wie kann man aber sicher sein, dass ein öffentlicher Schlüssel wirklich dem angenommenen Absender gehört?

Dazu gibt es mehrere Möglichkeiten:

- Man verifiziert den Schlüssel bei einer persönlichen Übergabe
- Man verifiziert den Schlüssel anhand seines *Fingerabdruckes*
- Man bekommt den Schlüssel von jemandem, dem man vertraut

In PGP gibt es eine Schlüsselverwaltung, die es erlaubt, den *Trust* in einen Schlüssel zu bestimmen.

Dazu gibt es in PGP das *Web of Trust*.

Das Konzept des Web of Trust:

- Schlüssel können ebenfalls beglaubigt (signiert) werden.
- Wenn jemand, dem man voll vertraut, den Schlüssel beglaubigt wird auch dem Schlüssel voll vertraut
- Manchen Schlüsseln wird nur zum Signieren vertraut

Umgang mit PGP

Zuerst muss man ein Schlüsselpaar erzeugen.

Dazu sollte man zuerst ein Directory `.pgp` anlegen:

```
> mkdir $HOME/.pgp
```

Danach wird mit dem `pgp` Programm ein Schlüsselpaar erzeugt:

```
> pgp -kg
```

Danach fragt `pgp` sehr ausführlich nach zusätzlichen Parametern:

- Der Länge des Schlüssels (512, 768, 1024 Bit)
- Der User-ID des Schlüssels (Real-Name `<email>`)
- Einer Passphrase für den privaten Schlüssel
- „Zufälligen Daten“ (Tastendrücke)

Bei PGP V5 ist der Vorgang ähnlich, aber anders:-)

Zum einen heisst das Programm `pgpk`, zum anderen hat man hier mehr Auswahl.

```
> mkdir .pgp
> pgpk -g
```

Zusätzlich zu den Daten von PGP V2 kann man nun:

- Einen anderen Verschlüsselalgorithmus angeben: DSS/DH statt RSA
- Eine Zeitspanne für die Gültigkeit des Schlüssels angeben
- Einen Key-Server angeben, an den der öffentliche Schlüssel gesandt wird

Von PGP V5 erzeugte DSS/DH Schlüssel können *nicht* von PGP V2 benutzt werden.

Umgekehrt sollte es schon gehen.

PGP hat nun die Schlüssel in den Dateien

- `secring.pgp` (bzw. `secring.pkr`) für den privaten Schlüssel
- `pubring.pgp` (bzw. `pubring.pkr`) für den öffentlichen Schlüssel

abgelegt.

Der private Schlüssel wird natürlich nicht im Klartext abgelegt, sondern ist wieder mit der Passphrase verschlüsselt in der Datei gespeichert.

Vorsicht: Verliert man diese Dateien, oder vergisst man die Passphrase, ist der Schlüssel nicht mehr benutzbar.

Um eine Nachricht für eine Reihe von Empfängern zu verschlüsseln, gibt man PGP die Encrypt-Option und die User-IDs der Empfänger, nebst File:

```
> pgp -e LoesungUebung Thesing "schellen@cs.uni-sb.de"
```

Die erzeugt dann ein File `LoesungUebung.pgp`, das den verschlüsselten Text enthält.

Dieses File kann nur noch von den beiden Empfängern dekodiert werden (nicht mehr vom Sender).

Man kann die Ausgabe natürlich auch umleiten:

```
> pgp -e LoesungUebung -o mail.crypt Thesing "schellen@cs.uni-sb.de"
```


Eine so erzeugte verschlüsselte Datei enthält Binärdaten.

Sie kann also nicht ohne weiteres per Email verschickt werden.

Um die Angelegenheit zu vereinfachen, bietet PGP an, die Datei als ASCII abzulegen (ASCII Armor):

```
> pgp -ea LoesungUebung Schellenbach
```

Der Inhalt sieht dann etwa so aus:


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

hEwD+9eKD6yNED0BAf9NsbZufLV0B0kqU/S0upITz+AHngnNGlANxow5NNchs6Xx
hF6x0utYw3EtWovyDLJDCfMTGVMRX9iGL6cET53mpgAAADYlSjUXGbVnWDk0qejV
ZHYwAkChFbuKehVv5Y+m/FJBLGS57696LGi7qgfFJK0aMcCL+06Sw5s=
=yv5Z
-----END PGP MESSAGE-----
```

Will man die Originaldatei nach dem Verschlüsseln sofort löschen, gibt es auch hierzu eine Option in PGP (wipeout):

```
> pgp -eaw LoesungUebung Schellenbach
```

Mit PGP V5 ist die Angelegenheit etwas anders:

```
> pgpe -aw LoesungUebung -r Schellenbach -r Thesing -o mail
```

Entschlüsseln

Das Entschlüsseln ist mit PGP einfach:

```
> pgp mail.crypt
```

bzw.

```
> pgp mail.crypt -o foobar
```


Auch hier ist PGP V5 etwas anders:

```
> pgpv mail.crypt -o foobar
```

Bei beiden Versionen wird nach der Passphrase des privaten Keys gefragt.

Wird kein Ausgabe-File gegeben, wird der Ausgabename aus dem Eingabefilenamen bestimmt.

Vor dem Überschreiben existierender Files wird in diesem Fall gewarnt.

Signieren

Um eine Nachricht zu signieren, kann man beim Verschlüsseln die `-s` Option geben:

```
> pgp -esa foobar Thesing
```

Danach muss man seine Passphrase angeben (Signieren braucht den privaten Schlüssel).

Beim Entschlüsseln weist PGP dann auf die vorhandene Signatur hin und ob sie gültig ist.

PGP V5:

```
> pgpe -sa foobar -r Thesing
```


Will man eine Nachricht nicht verschlüsseln, sondern nur signieren:

```
> pgp -s foobar
> pgp -sa foobar -o mail.ascii
```

Dann kann jeder diese Mail wieder in Originalform holen.

Jeder, der den öffentlichen Schlüssel des Absenders kennt, kann die Signatur prüfen.

Zum Prüfen/Ansehen der Nachricht:

```
> pgp mail.ascii
```

PGP fragt nach dem öffentlichen Schlüssel des Absenders (wenn es ihn nicht kennt).

PGP V5:

```
> pgps -a foobar -o mail.ascii
```

Verifikation:

```
> pgpv mail.ascii
```

Key Management

Zur Verwaltung von Schlüsseln kennt PGP eine Reihe von Kommandos:

- Neue öffentliche Schlüssel aufnehmen
- Schlüssel löschen
- Parameter editieren
- Fingerabdruck ausgeben

Öffentliche Schlüssel werden als sog. Key-Files weitergegeben.

Um einen Schlüssel (oder mehrere) aus einen Key-File zu den bekannten Schlüsseln hinzuzufügen:

```
> pgp -ka KeyFile
```

Um einen Schlüssel mit einer bestimmten ID zu löschen:

```
> pgp -kr Thesing
```

Um den eigenen öffentlichen Schlüssel in ein KeyFile zu schreiben:

```
> pgp -kxa Thesing KeyFile
```

Zum Ansehen von Schlüsseln:

```
> pgp -kv Schellenbach
> pgp -kvv Schellenbach
> pgp -kv
```

Zum Ausgeben des Fingerabdruckes:

```
> pgp -kvc Schellenbach
```

Der Fingerabdruck ist eine einfache Möglichkeit, erhaltenen Schlüssel zur verifizieren.

Um die Signaturen von Schlüsseln zu sehen/prüfen:

```
> pgp -kc Schellenbach
> pgp -kc
```

Zum Editieren der Parameter eines Schlüssels (Passphrase, Trust Parameter):

```
> pgp -ke Thesing
```

Um nun einen fremden öffentlichen Schlüssel zu signieren:

```
> pgp -ks Schellenbach
```


Version für PGP V5 (pgpk ist da verantwortlich):

Schlüssel aus Key-File hinzufügen:

```
> pgpk -a KeyFile
```

Signaturen prüfen:

```
> pgpk -c Schellenbach
```

Einen Schlüssel aktivieren/deaktivieren:

```
> pgpk  -d Thesing
```

Zum Editieren eines Schlüssels:

```
> pgpk -e Thesing
```

Ausgabe von Schlüsselparametern:

```
> pgpk -l Thesing
> pgpk -ll Schellenbach
> pgpk -ll
```

Einen Schlüssel löschen:

```
> pgpk -r Schellenbach
```

Einen Schlüssel signieren:

```
> pgpk -s Thesing
```

Und einen Schlüssel extrahieren:

```
> pgpk -x Thesing
```