

GNU Privacy Guard (GPG)

Matthias Schmidt
schmidt@giessen.ccc.de

Juni 2002

Warum überhaupt verschlüsseln?

- **Vertraulichkeit**

- E-Mail so sicher wie eine Postkarte
- Mailserver Administratoren könnten die E-Mails lesen
- Mails können auf dem Weg zum Ziel abgefangen werden
- Unbefugtes Lesen/Öffnen verhindern
- Elektronisches Filtern der E-Mails (ECHELON) verhindern
- Nach GG Artikel 10, Absatz 1 gilt das Briefgeheimniss

- **Integrität**

- Die Daten sollen auch von der Person kommen, von der sie erwartet werden

- **Authenzität**

- Fälschen einer Absenderadresse problemlos möglich
- Der Empfänger will auch die Daten haben, die abgesendet wurden
- Bei offizieller Korrespondenz oder bei Geschäften ist es wichtig den Sender eindeutig zu identifizieren

Was ist überhaupt GPG?

- Programm zum Ver- und Entschlüsseln von Daten
- Digitales Signieren von Daten ist möglich
- Ersatz für Philip Zimmermanns PGP
 - Voll zu PGP 5.x kompatibel
- RFC 2440 (OpenPGP) kompatibel

Warum gerade GPG?

- **OpenSource**

- Lizenziert unter der GNU GPL
- Frei von Patenten
- Frei von einschränkenden Lizenzbedingungen

- **Unabhängig vom Datenformat**

- E-Mails
- Textdateien
- Bilder
- Datenbanken
- Festplatten

- **Sicherheit**

- Keine Hintertüren
- Keine Generalschlüssel
- Nicht durch nationale Krypto-Ausfuhrbestimmungen beschränkt

Wie funktioniert GPG?

- **Public-Key Verschlüsselung**

- Es gibt einen *öffentlichen (public)* und einen *privaten (secret)* Schlüssel
- Den *secret key* speichert man auf einem "sicheren" Medium (Diskette, Festplatte, Memory-Stick, Chipkarte, ...)
- Den *public key* kann man verschicken oder z.B. auf öffentlichen Key-Servern deponieren
- Will z.B. Person A eine Nachricht an B versenden, verschlüsselt sie die Nachricht mit dem *public key* von B und sendet sie dann.
- B benutzt nun seinen *secret key* um die Nachricht wieder zu entschlüsseln.

- **Hybride Verschlüsselung**

- Verschlüsseln der eigentlichen Nachricht mit einem Sitzungsschlüssel
- Dieser wird wiederum mit dem *public key* des Empfängers verschlüsselt
- Der Empfänger entschlüsselt dann mit seinem *secret key* den Sitzungsschlüssel und kann dann die ursprüngliche Nachricht dekodieren

Sicherheit

Folgende Punkte sollten bei der Benutzung von GPG beachtet und bedacht werden ...

- **Schlüsselgenerierung**

- Die Wahl der Schlüssellänge des *private* und *public keys*
- Das Verfallsdatum der Schlüssel

- **Sicherheit**

- Auf einem Multiuser-System (UNIX, VMS) hat der *superuser* potentiellen Zugriff auf den keyring des Benutzers
- Das Mantra kann mit geeigneten Mittel mitgelesen werden oder durch Unachtsamkeit bekannt werden
- Aus Versehen wird der *secret key* verschickt
- Einsatz von GPG auf Windows-Plattformen

Erzeugen eines GPG-Schlüsselpaares

```
$ gpg --gen-key
gpg (GnuPG) 1.0.7; Copyright (C) 2002 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.
```

Please select what kind of key you want:

- (1) DSA and ElGamal (default)
- (2) DSA (sign only)
- (4) ElGamal (sign and encrypt)
- (5) RSA (sign only)

Your selection? 1

DSA keypair will have 1024 bits.

About to generate a new ELG-E keypair.

minimum keysize is 768 bits

default keysize is 1024 bits

highest suggested keysize is 2048 bits

What keysize do you want? (1024) 2048

Erzeugen eines GPG-Schlüsselpaares (2)

Please specify how long the key should be valid.

0 = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years

Key is valid for? (0) 0

Key does not expire at all

Is this correct (y/n)? y

You need a User-ID to identify your key; the software constructs the user id from Real Name, Comment and Email Address in this form:

"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Matthias Schmidt

Email address: schmidt@giessen.ccc.de

Comment:

You selected this USER-ID:

"Matthias Schmidt <schmidt@giessen.ccc.de>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? 0

Erzeugen eines GPG-Schlüsselpaares (3)

You need a Passphrase to protect your secret key.

Enter passphrase:

Repeat passphrase:

We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.

[...]

public and secret key created and signed.

key marked as ultimately trusted.

```
pub 1024D/B47BDCAB 2002-08-01 Matthias Schmidt <schmidt@giessen.ccc.de>  
    Key fingerprint = B472 A547 3E54 54F8 E818 4E19 6D85 CD1A B47B DCAB  
sub 2048g/6D2CD61C 2002-08-01  
$
```

Weiterführende Informationen

- **GnuPG Website** (<http://www.gnupg.org>)
 - Dokumentation (u.a. GNU-Handbuch zum Schutze der Privatsphäre)
 - Diverse Maillingslisten zu GPG
 - Download der aktuellsten Version
- **Projekt Ägypten Website** (<http://www.gnupg.org/aegypten/>)
 - Erweiterungen für Arbeitsplatzsysteme
- **Counterpane Systems** (<http://www.counterpane.com>)
 - Firma des Crypto-Gurus Bruce Schneier