GnuPG Einführung

E-Mails und Daten verschlüsseln

Emanuel Duss

2015-02-04

Was ist GnuPG

- Verschlüsselung von E-Mail und Daten
- Sichere Kommunikation
- End zu End Verschlüsselung

Schlüsselpaar generieren: Gute Zufallszahlen

Haveged füllt den Entropiepool für /dev/random:

```
$ cat /proc/sys/kernel/random/entropy_avail
35

$ sudo pacman -S haveged
$ sudo systemctl enable haveged
$ sudo systemctl start haveged

$ cat /proc/sys/kernel/random/entropy_avail
2436
```

Schlüsselpaar erstellen

Neues Schlüsselpaar generieren:

```
$ gpg --full-gen-key
```

- RSA and RSA (default): Algorithmus zum Verschlüsseln und Signieren
- 4096: Schlüsselgrösse wird 4096 Bit
- 2y: Der Schlüssel läuft in zwei Jahren automatisch ab
  - (Hinweis: Das Ablaufdatum kann aber jederzeit geändert werden. Falls man den Schlüssel aber verlieren sollte, ist man froh, wenn dieser automatisch nach einer gewissen Zeit ungültig wird)
- Rainer Zufall: Vorname und Name
- rainer.zufall@example.net: E-Mail Adresse

Schlüsselpaar anzeigen

Private Schlüssel anzeigen (zum Entschlüsseln und Signieren):

```
$ gpg -K
```

Öffentliche Schlüssel anzeigen (zum Verschlüsseln und Signatur überprüfen):

```
$ gpg -k
```

Zusätzliche User IDs

Zusätzliche User IDs für mehrere E-Mail Adressen.

```
$ gpg --edit-key 0xB1940999
gpg> adduid
Real name: Rainer Zufall
Email address: rainer.zufall@example.org
gpg> save
```

Keys publizieren

Keyserver aus Pool verwenden und immer diese verwenden:

```
$ vi ~/.gnupg/gpg.conf
keyserver hkps://hkps.pool.sks-keyservers.net
keyserver-options no-honor-keyserver-url
keyid-format 0xshort # Darstellung mit 0x Präfix
```

Key senden:

```
$ gpg --send-keys 0xB1940999
```

Public Key als Datei exportieren:

```
$ gpg --armor --export-options export-minimal --export 0xB1940999
```

Keys empfangen

Nach Key suchen:

```
$ gpg --search-key Rainer Zufall
```

Key herunterladen:

```
$ gpg --recv-key 0x23F00BA2
```

Fingerprint vergleichen:

```
$ gpg --fingerprint 0x23F00BA2
```

Backup erstellen

Alles sichern:

```
$ cp -r .gnupg/ /mnt/securedevice
```

Wiederrufszertifikat

```
$ gpg --gen-revoke --armor 0xB1940999 > 0xB1940999_revoke.asc
```

Private Key & Subkeys zum ausdrucken:

```
$ gpg --armor --export-secret-keys 0xB1940999 > 0xB1940999_private.asc
```

Nachricht verschlüsseln

```
$ gpg --encrypt --armor --recipient emanuel
Foo!
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2

hQIMA8yLbZkb4/puARAAqPGSQKHFFN3aiJCijz5asaL1XVWamzKFYJvvILnoHTTW
iB/nG7XWtzz+jtL360L4sscRRjmXg7YnxPR5Cl38WGb7RLsq/JJKXKL0lyIX1pI/
[...]
QAHGl0kxVr46pevgjSiz7uJM4c0IoooMt25u8hUhvUZ+joee6ay2LH9IsAg0EE+B
Bph9JBZ2lOVIJyIqSHUJJIY=
=c7Ek
-----END PGP MESSAGE-----
```

Nachricht entschlüsseln

```
$ gpg --decrypt --armor
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2

hQIMA8yLbZkb4/puARAAqPGSQKHFFN3aiJCijz5asaL1XVWamzKFYJvvILnoHTTW
iB/nG7XWtzz+jtL360L4sscRRjmXg7YnxPR5Cl38WGb7RLsq/JJKXKL0lyIX1pI/
[...]
QAHGl0kxVr46pevgjSiz7uJM4c0IoooMt25u8hUhvUZ+joee6ay2LH9IsAg0EE+B
Bph9JBZ2l0VIJyIqSHUJJIY=
=c7Ek
-----END PGP MESSAGE-----
gpg: encrypted with 4096-bit RSA key, ID 0x1BE3FA6E, created 2014-12-25
      "Emanuel Duss <emanuel.duss@gmail.com>"
Foo!
```

Nachricht signieren

```
$ echo Fnord! > message

$ gpg --detach-sign --armor message

$ cat message.asc
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v2

iQIcBAABCAAGBQJU0k/EAAoJEHau+5k1J4sODOIP/jaBgM6ReAlUnL9IZ0t5gXwu
vajGWUxdYEZN0Kn7A7Qgn5xq3GN17h2s0H7WMi4BxKOF8dSCReePSdxksoQ2a4Vp
[...]
CSCaJjBy9upyWNS9HcjovwG86H8qTf5pLjT+10FCu/z3rudj3TWre2U0atYMG9e0
3EGqvAA86lTLhVkawSbT
=zBHD
-----END PGP SIGNATURE-----
```

Signatur überprüfen

```
$ gpg --verify message.asc message
gpg: Signature made Wed 04 Feb 2015 05:57:00 PM CET
gpg:                using RSA key 0x35278B0E
gpg: Good signature from "Emanuel Duss <emanuel.duss@gmail.com>" [full]
gpg:                 aka "Emanuel Duss <eduss@hsr.ch>" [full]
```

Keys signieren

Key besorgen:

```
$ gpg --recv-key 0x23F00BA2
$ gpg --import < key.asc
```

Fingerprint anzeigen und vergleichen:

```
$ gpg --fingerprint 0x23F00BA2
```

Key signieren:

```
$ gpg --sign-key 0x23F00BA2
```

Auf Keyserver laden:

```
$ gpg --send-keys 0x23F00BA2
```

GnuPG mit Thunderbird: Enigmail

- Erweiterung Enigmail installieren
- Betreff wird nicht verschlüsselt / signiert
- PGP/MIME verwenden
- Entwürfe lokal speichern

Weitere Schritte

- Sicheres GnuPG Setup: Subkeys erstellen und Primary Key offline speichern:
  - `https://emanuelduss.ch/2015/01/sicheres-gnupg-setup-primary-key-offline-speichern/`
- Wir signieren gegenseitig unsere Keys

Referenzen

- OpenPGP Best Practices:
  - https://help.riseup.net/en/security/message-security/openpgp/best-practices
- Generating More Secure GPG Keys: A Step-by-Step Guide:
  - http://spin.atomicobject.com/2013/11/24/secure-gpg-keys-guide/

Fragen

?