GPG / GnuPG / PGP

- GPG / GnuPG / PGP Cheat-Sheet
- PGP public key von Keyserver holen und exportieren

GPG / GnuPG / PGP Cheat-Sheet

- GPG (bzw. gpg) ? OpenPGP encryption and signing tool
- GnuPG ? GNU Privacy Guard
- PGP ? Pretty Good Privacy

GPG-Tools und Verwaltung unter Linux

Keymanager / Grafische Frontends

- gpa ? GNU Privacy Assistant (für GTK)
- seahorse ? Frontend für GNOME
- kleopatra ? Frontend für KDE
- kgpg ? Frontend ebenfalls für KDE

GPG Basics

neuen Key erstellen

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

Keys anzeigen

```
# nur öffentliche Schlüssel anzeigen
$ gpg --list-keys

# nur private Schlüssel anzeigen
$ gpg --list-secret-keys
```

Keys exportieren

```
# öffentlichen Schlüssel exportieren:
$ gpg --export -a "Username/KeyID" > user-pub.asc

# privaten Schlüssel exportieren:
$ gpg --export-secret-key -a "Username/KeyID" > user-pub-sec.asc
```

Keys importieren

```
$ gpg --import user-pub.asc
$ gpg --import user-pub-sec.asc # es wird nicht zw. public/secret keys unterschieden
```

Keys löschen

```
$ gpg --delete-key "Username/KeyID"
$ gpg --delete-secret-key "Username/KeyID"
```

Fingerprints für den gesamten Schlüsselbund anzeigen

```
$ gpg --fingerprint
```

Trust-Database exportieren/importieren

```
# Export:
$ gpg --export-ownertrust > trust.txt

# Import:
gpg --import-ownertrust < trust.txt
```

Widerrufszertifikat erzeugen

Mit einem solchen Zertifikat lassen sich Schlüssel auf Keyservern für ungültig erklären, z.B. wenn man den privaten Schlüssel verloren oder sich einen neuen erstellt hat.

```
$ gpg --gen-revoke "Username/KeyID"
```

Passphrase eines Schlüssels ändern

```
$ gpg --edit-key "Username/KeyID" passwd
```

Vertrauensverhältnis festlegen

```
$ gpg --edit-key "Username/KeyID" trust
```

ICQ und GPG

Die ICQ-UIN als neue UserID an den eigenen Key hängen.

> Das ist nicht unbedingt Standard-Konform

1. zuerst die UID rausfinden, sofern nicht bekannt:

```
gpg --list-secret-keys
```

2. mit folgendem Befehl fügt man eine neue (nicht standard-konforme) UID an den Key an:

```
gpg --allow-freeform-uid --edit-key <keyid> adduid
```

3. das Ganze sieht dann etwa so aus:

```
gpg (GnuPG) 1.4.7; Copyright (C) 2006 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.

Secret key is available.

[...]

Real name: Hans Mustermann
Comment:
You selected this USER-ID:
    "Hans Mustermann"
```

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

You need a passphrase to unlock the secret key for
user: "Hans Mustermann <hans@mustermann.de>"

[...]

Command> quit
Save changes? (y/N) y
```

PGP public key von Keyserver holen und exportieren

Das folgende kleine Script lädt einen GPG-Pubkey von einem Keyserver herunter und exportiert ihn im ASCII-Format in eine Datei. Als Argument übergibt man ihm die ID des Keys (so wird dann auch die Datei benannt).

Der Key liegt danach ebenfalls im lokalen Keystore.

```
gpg-get-key.sh#!/usr/bin/env bash
#
# download a public key from a keyserver and write it ascii-armored to a file
#

KEYSERVER="keyserver.ubuntu.com"
GPGOPTS="--batch --quiet"

if [[ ${#} -eq 0 ]]; then
  echo "Usage: ${0} <keyid>"
  exit 1
fi

gpg ${GPGOPTS} --keyserver ${KEYSERVER} --recv ${1}

if [[ ${?} -ne 0 ]]; then
  echo "key '${1}' not found"
  exit 1
fi

gpg ${GPGOPTS} --export --armor -o ${1}.asc ${1}

if [[ -e ${1}.asc ]]; then
  echo "successfully exported the key to '${1}.asc'"
fi
```