GnuPG *Cheatsheet* written by Daniel Vogelbacher

Encrypt and decrypt

Encrypt a file with your key

```
gpg -er me@example.org diary.txt
```

-r encrypts the file for the user specified as argument. The output is written to *diary.txt.gpg*.

Encrypt a file for another person

```
gpg -er alice@example.org diary.txt
```

Encrypt a file with passphrase

```
gpg -c diary.txt
```

Decrypt a file

```
gpg -do diary.txt diary.txt.gpg
```

Encrypt and sign

Encrypt a file and sign with your key

```
gpg -esr alice@example.orge diary.txt
```

Sign and decrypt the file, output is written to *diary.txt.gpg*.

Decrypt and verify a file

```
gpg -do diary.txt diary.txt.gpg
```

Signing and verification

Signing compresses the file and append a signature.

Sign a file

```
gpg -s diary.txt
```

The signature is written to *diary.txt.gpg*.

Verify a signature

```
gpg --verify diary.txt.gpg
```

Verify a signature and restore the file

```
gpg -do diary.txt diary.txt.gpg
```

-d is for decryption, this works for signature files as well.

Special forms of signing

With *clearsign*, the output is wrapped in an ASCII-armored signature.

Clearsign a file

```
gpg --clearsign diary.txt
```

The signature is written to *diary.txt.asc*.

Verify a clearsign signature

```
gpg --verify diary.txt.asc
```

Verify a signature and restore the file

```
gpg -do diary.txt diary.txt.asc
```

-d is for decryption, this works for signature files as well.

Detached signatures are just signatures without any data

Sign a file using detached signature

```
gpg -o diary.txt.sig --detach-sig diary.txt
```

Verify a detached signature

```
gpg --verify diary.txt.sig diary.txt
```

Public key management

List all public keys in keyring

```
gpg -k
```

Export a public key

```
gpg -ao alice_pub.asc --export alice@example.org
```

Import a public key

```
gpg --import alice_pub.asc
```

Verify a public key fingerprint

```
gpg --fingerprint alice@example.org
```

Secret key management

Generate key pair (RSA,DSA, ECC like Curve 25519)

```
gpg --expert --full-gen-key
```

Using *expert mode* for key generation enables the choice of different key types.

Edit secret key

```
gpg --expert --edit-key me@example.org
```

Most interesting commands:

adduid	Create an additional user ID
revuid	Revoke a user ID or photographic user ID
primary	Flag the current user id as the primary one
addkey	Add a subkey to this key
expire	Change the key or subkey expiration time
passwd	Change the passphrase of the secret key
save	Save all changes to the keyrings and quit

List all secret keys

```
gpg -K
```

Generate revocation certificate

```
gpg -ao revoke-key.asc --gen-revoke A767285029E8882A
```

Deleting keys

```
gpg --delete-secret-key A767285029E8882A
gpg --delete-key A767285029E8882A
```

Make sure you have published a revocation certificate, if your key was uploaded to a key server!

Signing parties

Show the public key fingerprint

```
gpg --fingerprint 00AA11BB22CC33DD
```

Sign a public key

```
gpg --sign-key 00AA11BB22CC33DD
```

List keys with signatures

```
gpg --list-sigs
```

Generate fingerprint paper slips

```
gpg-key2latex --show-qrcode 00AA11BB22CC33DD
```

Key servers

Send public key to a keyserver

```
gpg --keyserver hkp://pgp.mit.edu --send-keys A767285029E8882A
```

Remember it's not possible to remove a key from any keyserver!

Get public key from a keyserver

```
gpg --keyserver hkp://pgp.mit.edu --recv-key A767285029E8882A
```

Search public key on a keyserver

```
gpg --keyserver hkp://pgp.mit.edu --search-keys Alice
```

Refresh all keys from keyserver

```
gpg --keyserver hkp://pgp.mit.edu --refresh-keys
```

Useful for updating a key with the latest signatures, user IDs, etc. Calling this with no arguments will refresh the entire keyring.

Fancy features

Export SSH compatible public key

```
gpg --export-ssh-key me@example.org
```

Export a key in the OpenSSH public key format. You need a subkey for authentication purposes *[A]*.

Get keygrip

```
gpg --list-secret-keys --with-keygrip
```

The keygrip can be put in *~/.gnupg/sshcontrol* to use GnuPG for ssh authentication.

Export subkeys

```
gpg -o my_subkey.gpg --export-secret-subkey F988A393A99DB2F3!
```

Export only one or more subkeys for a stripped keyring. Remember the required *!* at the end.

Update trust db

```
gpg --update-trustdb
```