DEVHINTS.IO Edit

# GnuPG cheatsheet



GnuPG is a complete and free implementation of the OpenPGP standard.

# Basics

## Exporting keys

```
gpg -o key.gpg --export <KEY ID>
```

**Export key in ASCII:**

```
gpg -o key.asc --armor --export <KEY ID>
```

**Note:** Omitting the `-o|--output` option will print the key to `stdout`.

## Importing keys

```
gpg --import key.gpg
gpg --import key.asc
```

Only merge updates for keys already in key-ring:

```
gpg --import key.asc --merge-options merge-only
```

## Managing your keyring

Generate a new key:

```
gpg --gen-key
# or, generate a new key with dialogs for all options
gpg --full-gen-key
```

List public keys:

```
gpg -k
gpg --list-keys
```

List secret keys:

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

## Using a keyserver

Import keys from keyserver:

```
gpg --receive-keys <KEY IDS>
```

Upload keys to keyserver:

```
gpg --send-keys <KEY IDS>
```

Request updates from keyserver for keys already in your keyring:

```
gpg --refresh-keys
```

Search keys from keyserver:

```
gpg --search-keys "<SEARCH STRING>"
```

Override keyserver from `~/.gnupg/gpg.conf`

```
gpg --keyserver <URL> ...
```

### Trusting a key

```
gpg --edit-key <KEY ID>
# In the interactive prompt:
gpg> sign
gpg> save
```

**NOTE:** You can use the owner's email or name (or part thereof) instead of the key ID for `--edit-key`

## Encrypting

### Public key encryption

This will produce an encrypted file, `secret.txt.gpg`, that can only be decrypted by the recipient:

```
gpg -e -o secret.txt.gpg -r <RECIPIENT> secret.txt
```

For `<RECIPIENT>` you can use their key ID, their email, or their name (or part thereof).

```
gpg -e -r <KEY ID> ...
gpg -e -r "Bez" ...
gpg -e -r "bezalelhermoso@gmail.com" ...
```

Specifying multiple recipients

```
gpg -e -r <RECIPIENT> -r <ANOTHER RECIPIENT> ... secret.txt
```

**NOTE**: Omitting `-o|--output` will produce an encrypted file named `<ORIGINAL FILENAME>.gpg` by default.

### Symmetric encryption

Encrypt file using a shared key. You will be prompted for a passphrase.

```
gpg --symmetric secret.txt
# or
gpg -c secret.txt
```

## Decrypting

### Decrypting a file

```
gpg -d -o secret.txt secret.txt.gpg
```

If the file is encrypted via symmetric encryption, you will be prompted for the passphrase.

**NOTE**: Omitting `-o|--output` will print the unencrypted contents to `stdout`

## Signing & Verifying

### Signing

```
gpg -o signed-file.txt.gpg -s file.txt
```

This can be used during encryption to also sign encrypted files:

```
gpg -s -o secret.txt.gpg \
  -r <RECIPIENT> secret.txt
```

### Verifying a signature

```
gpg --verify file.txt.gpg
```

### Viewing content of signed file

```
gpg -d signed-file.txt.gpg
```

## Miscellaneous

## Components

List all components:

```
gpgconf --list-components
```

Kill a component:

```
gpgconf --kill <COMPONENT> # i.e. gpgconf --kill dirmngr
```

Kill all components:

```
gpgconf --kill all
```

## Parsing keyring data

Use `--with-colons` to produce an output that can easily be parsed i.e. with `awk`, `grep`. Fields are colon-separated.

```
gpg -k --with-colons
```

Field Quick Reference:

| Field # | Description |
|---|---|
| 1 | Record type |
| 2 | Validity |
| 3 | Key length in bits |
| 4 | Public key algorithm |
| 5 | Key ID |
| 6 | Creation date |
| 7 | Expiry date |
| 8 | Certificate S/N, UID hash, trust signature info |
| 9 | Ownertrust |
| 10 | User ID |
| 11 | Signature class |
| 12 | Key capabilities |
| 13 | Issuer fingerprint |
| 14 | Flag field |
| 15 | S/N of token |
| 16 | Hash algorithm |
| 17 | Curve name |
| 18 | Compliance flags |
| 19 | Last update timestamp |
| 20 | Origin |

See GnuPG Details for more details.