GnuPG Encrypt and sign

CONCEPTS

Others encrypt to your public key and verify your signatures; only your secret key decrypts and signs. Guard the secret key.

Plaintext your file
↓ encrypt: recipient public key
Ciphertext .gpg / .asc
↓ decrypt: recipient secret key
Plaintext restored

A key ID can be an email, a name part, a long key ID or a full fingerprint. Prefer the fingerprint when it must be unambiguous.

KEYS

gpg --full-generate-key
Create a key pair interactively

gpg --quick-generate-key *<email>* **default default 2y**
Create a key with defaults, 2y expiry

gpg -k [*<id>***]**
List public keys

gpg -K [*<id>***]**
List secret keys

gpg --fingerprint *<id>*
Show the fingerprint

gpg --list-keys --keyid-format long
Show long key IDs

gpg --edit-key *<id>*
Edit a key in the key shell

gpg --delete-keys *<id>*
Remove a public key

gpg --delete-secret-keys *<id>*
Remove a secret key

IMPORT AND EXPORT

gpg --armor --export *<id>*
Print public key as text

gpg --export -o *<file>***.gpg** *<id>*
Export public key as binary

gpg --import *<file>*
Import a key file

IMPORT AND EXPORT (cont.)

gpg --armor --export-secret-keys *<id>* **>** *<file>***.asc**
Export the secret key

ENCRYPT AND DECRYPT

gpg -e -r *<id>* *<file>*
Encrypt to a recipient

gpg -ea -r *<id>* *<file>*
Encrypt as ASCII text (.asc)

gpg -c *<file>*
Encrypt with a passphrase only

gpg -d *<file>***.gpg**
Decrypt to stdout

gpg -o *<out>* **-d** *<file>***.gpg**
Decrypt into a file

gpg -e -r *<id1>* **-r** *<id2>* *<file>*
Encrypt to several recipients

gpg -e -r *<id>* **--output -** *<file>*
Encrypt stdin to stdout

SIGN AND VERIFY

gpg --detach-sign *<file>*
Write a separate .sig

gpg -ab *<file>*
Detached ASCII signature (.asc)

gpg --verify *<file>***.sig** *<file>*
Verify a detached signature

gpg --clearsign *<file>*
Sign text, keep it readable

gpg -s *<file>*
Sign and wrap the file

gpg --verify *<file>***.asc**
Verify a signed or clearsigned file

gpg -se -r *<id>* *<file>*
Sign and encrypt in one step

gpg -u *<id>* **--detach-sign** *<file>*
Sign with a specific key

KEYSERVERS AND TRUST

gpg --keyserver *<server>* **--send-keys** *<fpr>*
Upload your public key

KEYSERVERS AND TRUST (cont.)

gpg --keyserver *<server>* **--recv-keys** *<fpr>*
Fetch a key by fingerprint

gpg --search-keys *<email>*
Search keyservers

gpg --locate-keys *<email>*
Find a key via WKD/DANE

BACKUP AND REVOCATION

gpg --gen-revoke *<id>*
Create a revocation certificate

gpg -o *<file>***.rev --gen-revoke** *<id>*
Write the certificate to a file

gpg --import *<file>***.rev**
Revoke a key with the certificate

gpg --export-ownertrust > *<file>*
Back up trust settings

gpg --export-secret-keys --armor > *<file>***.asc**
Back up all secret keys

Keep an offline copy of the secret key and the revocation certificate; losing both means you cannot decrypt or retire the key.

AGENT AND CONFIG

gpgconf --kill gpg-agent
Stop the agent (restart on next use)

gpgconf --launch gpg-agent
Start the agent

gpg-connect-agent reloadagent /bye
Reload agent config

export GPG_TTY=$(tty)
Let pinentry find your terminal

RECIPES

Encrypt for a colleague

```
gpg --locate-keys <email>
gpg --fingerprint <email>
gpg -ea -r <email> -o <file>.asc
<file>
```

Sign a release and verify it

```
gpg -ab <file>.tar.gz
gpg --verify <file>.tar.gz.asc
<file>.tar.gz
```