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

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 --edit-key <id>`
Edit a key in the key shell

IMPORT AND EXPORT

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

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

`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)

ENCRYPT AND DECRYPT (cont.)

`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

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

KEYSERVERS AND TRUST

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

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

> 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)

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