

Initializing

Interactively generate my key: `gpg --gen-key`

List all public keys I have: `gpg --list-keys`

Generate a “violated key” message: `gpg --output revokedkey.asc --gen-revoke 0xkey_num`

Encrypting

Encrypt using their public key (and sign it):

```
gpg --recipient their_username --sign --encrypt file
```

Encrypt using a passphrase: `gpg --symmetric filename`

Decrypting

Decrypt a file: `gpg --decrypt filename`

Decrypt STDIN: `gpg --decrypt`

Signing

Sign a message: `gpg --clearsign filename`

Sign a message in separate file: `gpg --detach-sign filename`

Verifying: `gpg --verify signedfile`

```
gpg --verify signedfile.sig signedfile
```

Keyrings

Export a/my public key: `gpg --armor --export username > keyname.asc`

Import someone else’s public key: `gpg --import user-key.asc`

Get public key from a public key server:

```
gpg --recv-keys --keyserver wwwkeys.gpg.net user@some-email.com
```

Send a/my public key to a public key server:

```
gpg --send-keys --keyserver wwwkeys.gpg.net user@some-email.com
```

Approving

Verifying that someone else’s key is correct. Typically involves talking to them directly to verify that the public key you have is the one they sent.

Command line: `gpg --sign-key keyID`

Interactive, more powerful/flexible: `gpg --edit-key keyID`

Restricting to ASCII

```
gpg --armor --recipient username --sign --encrypt filename
```

```
gpg --armor --symmetric filename
```

```
gpg --armor --sign message
```