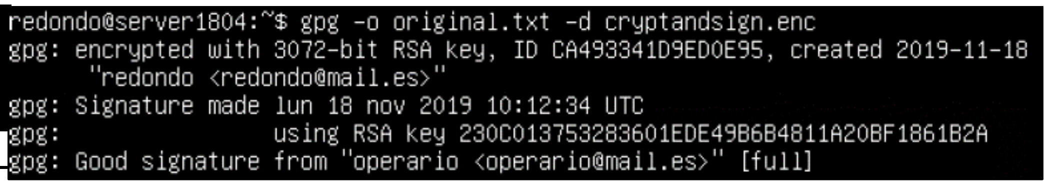
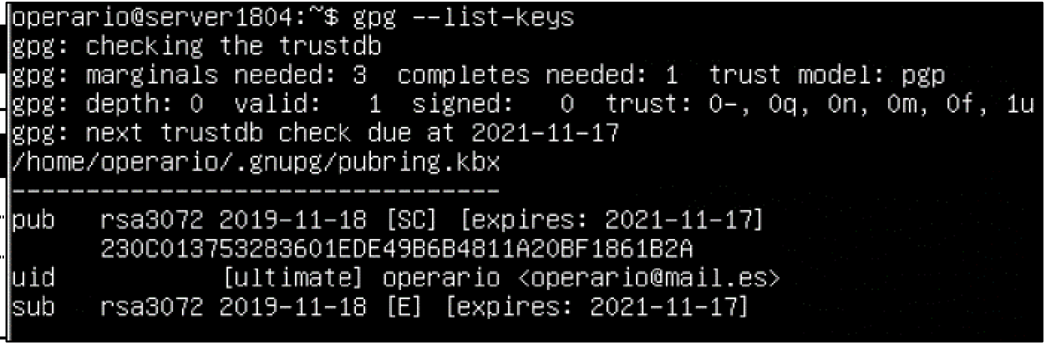
gpg (GnuPG) 2.2.20 Cheatsheet (ingenieriainformatica.uniovi.es)

Multipurpose GPL cipher/hash/pubkey software

https://gnupg.org/



```
redondo@server1804:~$ gpg -o original.txt -d cryptandsign.enc
gpg: encrypted with 3072-bit RSA key, ID CA493341D9ED0E95, created 2019-11-18
      "redondo <redondo@mail.es>"
gpg: Signature made lun 18 nov 2019 10:12:34 UTC
gpg:                using RSA key 230C013753283601EDE49B6B4811A20BF1861B2A
gpg: Good signature from "operario <operario@mail.es>" [full]
```



```
operario@server1804:~$ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2021-11-17
/home/operario/.gnupg/pubring.kbx
---------------------------------
pub   rsa3072 2019-11-18 [SC] [expires: 2021-11-17]
      230C013753283601EDE49B6B4811A20BF1861B2A
uid           [ultimate] operario <operario@mail.es>
sub   rsa3072 2019-11-18 [E] [expires: 2021-11-17]
```

GENERAL USAGE

`gpg [options] [files]`

NOTES

Sign, check, encrypt or decrypt

Default operation depends on the input data

Also supports the following compression algorithms: No compression, ZIP, ZLIB, BZIP2

OPTIONS

Symmetric encryption

Supported cipher algorithms: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH, CAMELLIA128, CAMELLIA192, CAMELLIA256. Use ***--cipher-algo*** *option to choose one*

-c, --symmetric: encryption only with symmetric cipher

Signatures

Supported hash algorithms: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224

--clear-sign: make a clear text signature

--verify: verify a signature

-b, --detach-sign: make a detached signature

-s, --sign: make a signature

Asymmetric encryption

Supported public key algorithms: RSA, ELG, DSA, ECDH, ECDSA, EDDSA

-d, --decrypt: decrypt data (default)

-e, --encrypt: encrypt data

Public/private key handling (I)

--card-status: print the card status

--change-passphrase: change a passphrase

--change-pin: change a card's PIN

--check-signatures: list and check key signatures

--delete-keys: remove keys from the public keyring

--delete-secret-keys: remove keys from the secret keyring

--edit-card: change data on a card

--edit-key: sign or edit a key

--export: export keys

--fingerprint: list keys and fingerprints

EXAMPLES

Sign and encrypt for user Bob: *gpg -se -r Bob [file]*

Make a clear text signature: *gpg --clear-sign [file]*

Make a detached signature: *gpg --detach-sign [file]*

Show keys: gpg *--list-keys [names]*

Show fingerprints: *gpg --fingerprint [names]*

Cipher a file using a strong symmetric key algorithm: *gpg --symmetric --cipher-algo AES256 -c message.txt*

Public/private key handling (II)

--full-generate-key: full featured key pair generation

--generate-key: generate a new key pair

--generate-revocation: generate a revocation certificate

--import: import/merge keys

--list-signatures: list keys and signatures

--lsign-key: sign a key locally

--print-md: print message digests

--quick-add-uid: quickly add a new user-id

--quick-generate-key: quickly generate a new key pair

--quick-lsign-key: quickly sign a key locally

--quick-revoke-uid: quickly revoke a user-id

--quick-set-expire: quickly set a new expiration date

--quick-sign-key: quickly sign a key

--receive-keys: import keys from a keyserver

--refresh-keys: update all keys from a keyserver

--search-keys: search for keys on a keyserver

--send-keys: export keys to a keyserver

--server: run in server mode

--sign-key: sign a key

--tofu-policy VALUE: set the TOFU policy for a key

--update-trustdb: update the trust database

-k, --list-keys: list keys

-K, --list-secret-keys: list secret keys

Miscellaneous options

--openpgp: use strict OpenPGP behavior

--textmode: use canonical text mode

-a, --armor: create ascii armored output

-i, --interactive: prompt before overwriting

-n, --dry-run: do not make any changes

-o, --output FILE: write output to FILE

Cipher files with a user-friendly Data Format: *gpg --armor --symmetric --cipher-algo AES256 –c message.txt*	**-r, --recipient USER-ID**: encrypt for USER-ID
Decipher a file that used symmetric encryption: *gpg --decrypt --output=dmessage.txt message.txt.gpg*	**-u, --local-user USER-ID**: use USER-ID to sign or decrypt
Clear signature for a file: *gpg --clearsign file.txt*	**-v, --verbose**: verbose
Assymetric encryption for a particular user (redondo): *gpg -o file_for_redondo.gpg -e -r redondo file.txt*	**-z N**: set compress level to N (0 disables)
Decryption of asymmetrically encrypted text: *gpg -o file.txt -d file.txt.gpg*	

by José Manuel Redondo López