Encryption and Decryption of a File using GPG Encryption Tool

An Instance of using GPG to encrypt a file before transmission and decrypting the same by the recipient.

GPG – Gnu Private Guard

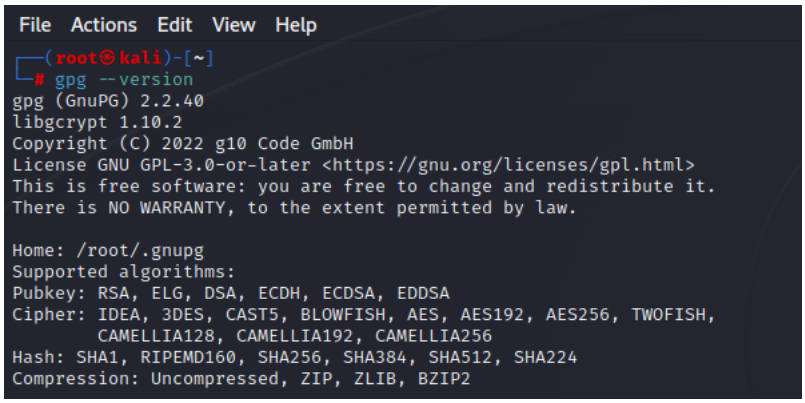
- Used for encryption (symmetric and asymmetric)
- Uses the concept of asymmetric encryption

Installing GPG - Ubuntu/any Debian based distributions

>sudo apt install gnupg

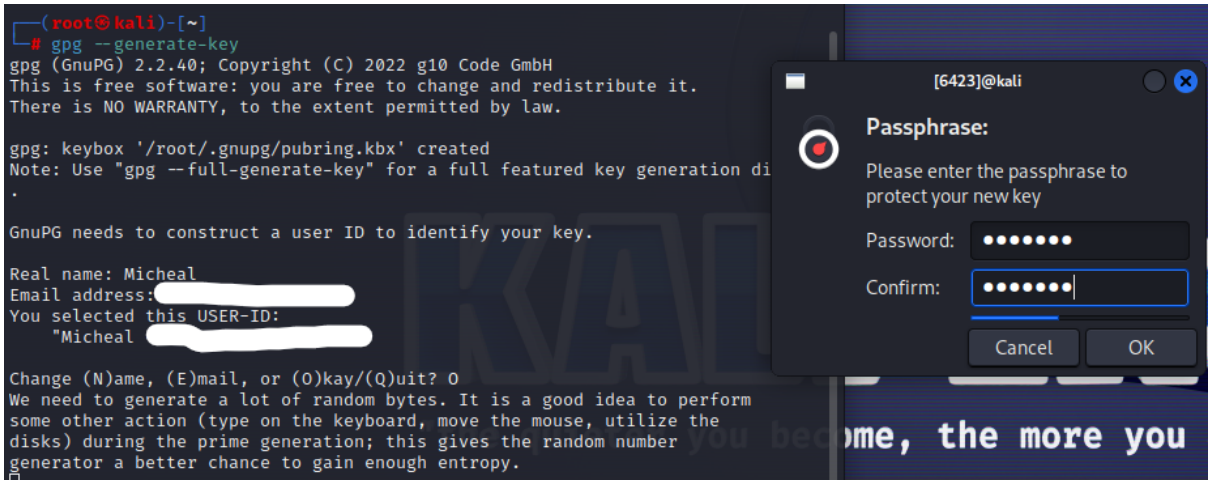
#Verify GPG Installation

>gpg --version



Generate a new key with default config

> gpg –-generate-key



```
Change (N)ame, (E)mail, or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/root/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/EFFB57DF
[redacted].rev'
public and secret key created and signed.

pub   rsa3072 2024-08-27 [SC] [expires: 2026-08-27]
      [redacted]
uid                      Micheal [redacted]
sub   rsa3072 2024-08-27 [E] [expires: 2026-08-27]
```

NB:To generate a new key with your own configuration:

> gpg –-full-generate-key

List all GPG keys

> gpg –-list-keys

```
┌──(root㉿kali)-[~]
└─# 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 2026-08-27
/root/.gnupg/pubring.kbx
------------------------
pub   rsa3072 2024-08-27 [SC] [expires: 2026-08-27]
      [redacted]
uid           [ultimate] Micheal <[redacted]>
sub   rsa3072 2024-08-27 [E] [expires: 2026-08-27]
```

NB: List all GPG private keypairs

> gpg –-list-secret-keys

Export a public key in txt file format

> gpg –-armor –-export –-output filename.txt email-id

```
┌──(root㉿kali)-[~]
└─# cd Downloads

┌──(root㉿kali)-[~/Downloads]
└─# gpg --armour --export --output pubkey_v1.txt [redacted].com

┌──(root㉿kali)-[~/Downloads]
└─# ls
[redacted]
[redacted]  pubkey_v1.txt  [redacted]
```

```
┌──(root㉿kali)-[~/Downloads]
└─# cat pubkey_v1.txt
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNBGbNyYUBDACrREyP9CuB0X9iNARiqXOeB8i4sfd6/SmT2X7vqBzkVEr1qBsX
tsavQyvg8Gjalm0joydDb5kkfmyOxCMGCEtppzAY1ntH8Z1P2lKr5PU4V/3PQc84
DvAX9bxF4NKNCv7kNGldpOcaJtmULmAVEaf+Sqmiq142h658qIZwdVveYvocnSD+
fL2Riu8otjFscJIHBLeTAKRMqtfm5wNmg4GZ1Afdr1QaBO0bFCrHg3eSip+g9aOg
TIdO0p3P7kOioq8TtfBGWk1MaJp3XBje8sQIrzZQOTstd9B7sRRw7vvy1poCbBXG
prtsbP5E6jVzCdiBAkUbjQSaqdSPZ1cQWTqTg428xIBgcZMecZ3×0/aXqYtcMYik
9GSrLs565O4qU80ZR4o0J+BAvxPVejtewBCldtnHedVoHJpB44+F0PAkEH7IpAVR
XZ5TvBI5gcWeYnp+wPF3jLCGw/E1EOHmWjywQ+FU0Bhvla0xl8eJ9gaIdM01MBN4
epWGz9/6VhOytOsAEQEAAbQcTWljaGVhbCA8b3RhZDJtaWNAeWFob28uY29tPokB
1AQTAQoAPhYhBO/7V9/Kanz18txSed2Oy0Kdd/GXBQJmzcmFAhsDBQkDwmcABQsJ
CAcCBhUKCQgLAgQWAgMBAh4BAheAAAoJEN2Oy0Kdd/GXjGYL/RWBcI9mhR7Cny1n
vTsOqF+Sw9+PZpwKw6n4jeLTNJvEcM50RGQpx9kBG5E3kaYHHNr2/TXlU9QBL0ea
i7ZWQ+OjBGr0R1aEIjjsITzCmWLV2Puik6MLpRqGUHsn5qC08DZEP09UQ1mLg8Ae
DFK7HCnoqH2G7lhA6uup3UhDah+gxUIArKHMjHnC3U8ZCwWlhGk+F+t504vl+Iav
```

Encrypt a file/document (in a folder with public key)

> gpg --output encrypted-file.gpg --encrypt --recipient email-id unencrypted-file.pdf

```
┌──(root㉿kali)-[~/Downloads/encryption-project]
└─# ls
pubkey_v1.txt  sample-file.txt

┌──(root㉿kali)-[~/Downloads/encryption-project]
└─# cat sample-file.txt
This is a sample file with dummy text written.

The contents in this file need to be encrypted for testing purpose.

┌──(root㉿kali)-[~/Downloads/encryption-project]
└─# gpg --output sample-file-encrypted.gpg --encrypt --recipient [redacted]
com sample-file.txt

┌──(root㉿kali)-[~/Downloads/encryption-project]
└─# ls
pubkey_v1.txt  sample-file-encrypted.gpg  sample-file.txt

┌──(root㉿kali)-[~/Downloads/encryption-project]
└─# cat sample-file-encrypted.gpg
◆◆◆y◆gC◆◆◆
          ◆5◆n_◆◆◆O◆◆◆◆ C◆Lw◆~x◆rP◆¦◆◆S=◆I◆^◆3◆SY^◆<q▒yOJ6◆O
                                                    ◆V◆◆◆◆"◆E(◆◆◆◆r<◆
?ꞃV◆N◆.@◆G,◆◆\◆◆"u◆◆}◆◆_lfwj◆)c◆)◆
T◆◆◆4◆◆◆◆Bd◆g◆◆ψ◆V{◆◆◆uR◆◆◆◆5◆b◆◆◆,◆d◆◆◆◆:◆]]@<▒8◆◆h◆.◆A◆U◆▒◆◆◆&\V◆◆◆
                ◆◆◆Ö◆◆◆s◆4◆/&◆◆.◆w,◆<+5◆
◆◆◆◆◆▒◆oH◆◆◆t◆◆◆◆◆,◆3◆◆◆d◆hv◆◆-◆2'◆J◆]◆9◆◆asu9◆◆Ć       ,R
                                                  !◆▯e!◆◆◆-A98T◆
◆:~◆gm◆◆Y◆◆◆◆[◆◆\◆x◆◆◆◆pOyP7◆◆◆[◆◆◆ps◆◆H◆O◆@A◆Cz◆T◆◆}$p<Z`◆;◆◆◆◆◆
                                                          ◆◆◆◆
9◆◆◆z◆=◆◆8◆◆◆◆8UE$◆
◆*/'◆,◆a]◆
◆\ᵵ{x◆6◆◆◆◆◆Z-◆q◆◆:◆◆BkU◆◆I◆◆{◆jl◆,t◆◆8
◆◆cDm&A)◆◆◆Ǩ!◆V◆i◆◆◆◆◆Xć◆?MH◆o◆▒        ◆J◆
```

Decrypt the file/document using the passphrase of the encryption key

> gpg --output unencrypted-file1.pdf --decrypt encrypted-file.gpg