

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
```

```
File Actions Edit View Help
(root@kali)~]
# gpg --version
gpg (GnuPG) 2.2.40
libgcrypt 1.10.2
Copyright (C) 2022 g10 Code GmbH
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /root/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

Generate a new key with default config

```
> gpg --generate-key
```

The screenshot shows the terminal output of the `gpg --generate-key` command. It displays the GPG version (2.2.40), supported algorithms, and a message indicating that a keybox has been created. It then prompts for a user ID, where 'Micheal' is entered. A dialog box titled 'Passphrase:' is overlaid on the terminal, asking for a passphrase to protect the new key. The dialog has fields for 'Password:' and 'Confirm:', both filled with dots, and 'Cancel' and 'OK' buttons.

```
(root@kali)~]
# gpg --generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: keybox '/root/.gnupg/pubring.kbx' created
Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Micheal
Email address: [redacted]
You selected this USER-ID:
"Micheal [redacted]"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? 0
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.
```

```

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/EFB57DF
[redacted].rev'
public and secret key created and signed.

pub   rsa3072 2024-08-27 [SC] [expires: 2026-08-27]
      [redacted]
uid    [redacted] 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    [redacted] [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 --armor --export --output pubkey_v1.txt [redacted].com
(root@kali)~/Downloads# ls
[redacted]
pubkey_v1.txt

```

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

mQGNBGBNyYUBDACrREyP9CuB0X9iNARIqX0eB8i4sfd6/SmT2X7vqBzkVer1qBsX
tsavQyvg8Gjalm0joydb5kkfmy0xCMGCEtppzAY1nth8Z1P2lkr5PU4V/3PQc84
DvAX9bxF4NKNcv7kN6ldp0caJtmULmAVEaf+Sqmiq142h658qIZwdVveYvocnSD+
fL2Riu8otjFscJIHBLtAKRMqt5wNmG4GZ1Afdr1QaB00bFCrHg3eSip+g9aOg
TId00p3P7k0ioq8TtfBGWk1MaJp3XBje8sQIrzZQ0Tstd9B7sRRw7vvy1poCbBXG
prtsbP5E6jVzCdiBAKUbJQSaqdSPZ1cQWTqTg428xIBgcZMecZ3x0/aXqYtcMYik
9GSrLs56504qU80ZR4o0J+BAvxpVeJtewBCLdtnHedVoHJpB44+F0PAKEH7IpAVR
XZ5TvBI5gcWeYnp+wPF3jLCGw/E1EOHmWjyWQ+FU0Bhvla0x18eJ9gaIdM01MBN4
epWGz9/6Vh0yt0sAEQEAAAbQcTWLjaGVhbCA8b3RhZDZtaWNAeWZob28uY29tPokB
1AQTAQoAPhYhB0/7V9/Kanz18txSed20y0Kdd/GXBQJmzcmFAhsDBQkDwmcABQsJ
CACCbHUKCQGLagQWAgMBAh4BAheAAAJEN20y0Kdd/GXjGYL/RWBCI9mhr7Cny1n
vTs0qF+S9w+PZpwKw6n4jeLTNjvEcM50RGQpx9kBG5E3kaYHHNr2/TXLU9QBL0ea
i7ZWQ+OjBG0R1aEIjjsITzCmWLV2Puik6MLpRqGUHsn5qC08DZEP09UQ1mLg8Ae
DFK7HCnoqH2G7lhA6uup3Uhdah+gXUIArKHMjHnCU8ZCwWlhGk+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.

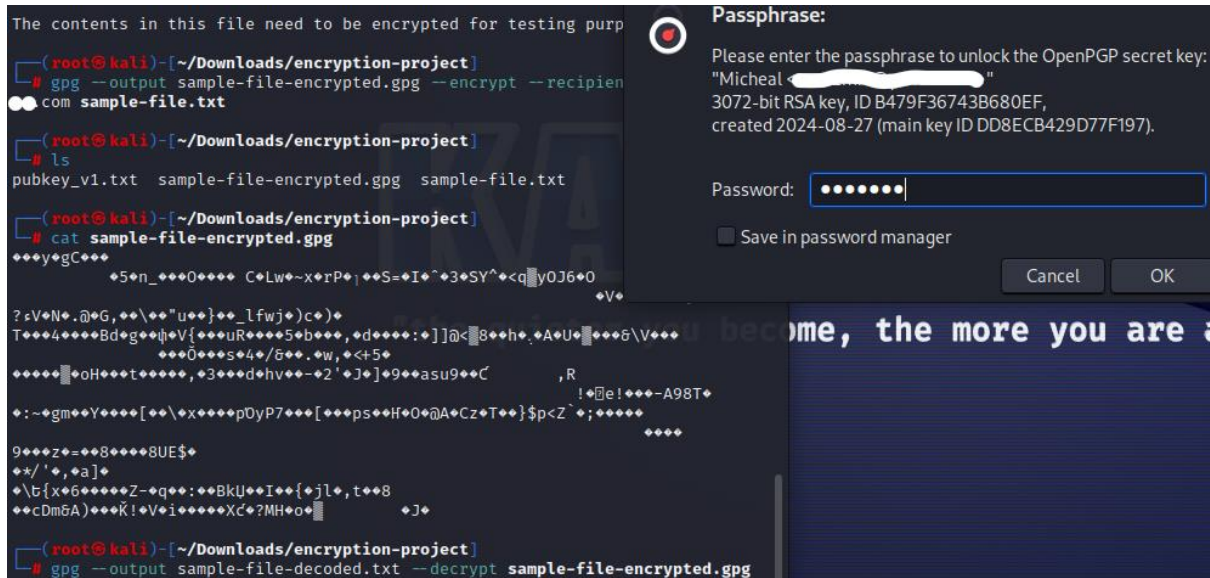
(root@kali)~[~/Downloads/encryption-project]
# gpg --output sample-file-encrypted.gpg --encrypt --recipient [redacted]
com sample-file.txt

(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
***yGCG***
+5+n_++0++++ C+Lw+-x+rP+;+S=+I+^+3+SY^+<q|y0J6+0
+V++++"E(++++r<+
?+V+N+.+@+G,+*+\"u++}++_lfwj+)c+)+
T++++4++++Bd+g+;+V{+uR++++5+b+++,+d++++:}]@<|8+eh+.+A+U+|++++6|V+++
++++0++++s+4+/5+++*w,+<+5+
+++++|+oH++++t++++,+3+++d+hv+-+2'+J+]9++++asu9+++C',R
+:~+gm++Y++++[+*+\\+x++++pDYP7++++[+*+ps++H+0+@A+Cz+T+}$p<Z`+;++++
++++
9+++z+=++8++++8UE$+
+*+'+*,+a)+
+\\t{x+6++++Z-+q++:++BKU++I++{+j|+,t++8
++cDm6A)+++K!+V+i++++Xc+?MH+o+|++J+
```

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

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



```

The contents in this file need to be encrypted for testing purp
(root@kali)~/Downloads/encryption-project
# gpg --output sample-file-encrypted.gpg --encrypt --recipient
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
***ygC***
+5+n_++0+++ C+Lw+-x+rP+;+S=+I+^+3+SY^<qY0J6+0
+V+
?zV+N+.@+G,++\++"u++}+_lfwj+c)+
T+++4++++Bd+g++h+V{+++uR++++5+b+++ ,d++++:}]@<8+h+.A+U+ +++G\V+++
+++0++++s+4+/G++ ,w, <+5+
+++++hO+++++ ,+3+++d+hv+-+2'+J+]*9++asu9++C ,R
+:~+gm++Y++++[++\x++++pDyP7+++[+++ps++H+0+@A+Cz+T++}$p<Z'+;++++
++++
9+++z+=++8++++8UE$+
+*/'+,a)+
+ \b{x+6++++Z-+q++:++BkU++I++{+jl+,t++8
++cDm6A)+++K!+V+i++++Xc+?MH+o+ +J+
(root@kali)~/Downloads/encryption-project
# gpg --output sample-file-decoded.txt --decrypt sample-file-encrypted.gpg
gpg: encrypted with 3072-bit RSA key, ID B479F36743B680EF, created 2024-08-27
"Micheal <[redacted]>"
the quieter you become, the more you are able to hear

```

```

(root@kali)~/Downloads/encryption-project
# gpg --output sample-file-decoded.txt --decrypt sample-file-encrypted.gpg
gpg: encrypted with 3072-bit RSA key, ID B479F36743B680EF, created 2024-08-27
"Micheal <[redacted]>"
the quieter you become, the more you are able to hear
(root@kali)~/Downloads/encryption-project
# ls
pubkey_v1.txt sample-file-encrypted.gpg
sample-file-decoded.txt sample-file.txt
(root@kali)~/Downloads/encryption-project
# cat sample-file-decoded.txt
This is a sample file with dummy text written.

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

```