***Pretty Good Privacy (PGP) is an encryption program that delivers cryptographic privacy and authentication for data communication. PGP is used for signing, encrypting, and decrypting texts, e-mails, files, directories, and whole disk partitions and to increase the security of e-mail communications. Phil Zimmermann developed PGP in 1991. PGP has been evolved throughout time into different formats. Similar software to PGP now follows the OpenPGP, an open standard of PGP encryption software, standard (RFC 4880) for encrypting and decrypting data. PGP is the backbone of Open PGP that allows PGP to be used in software that is typically free to the public. The term "Open PGP" is often applied to tools, features, or solutions that support open-source PGP encryption technology. GPG, or GnuPG, stands for GNU Privacy Guard. GPG is a different implementation of the Open PGP standard. GnuPG, popularly known as GPG, is an extremely versatile tool, being widely used as the industry standard for encryption of things like emails, messages, files, or just anything you need to send to someone securely. PGP is a proprietary solution owned by Symantec, and GPG (also known as GnuPG) is an open-source standard. Functionally, each format is virtually identical. It’s easy to get started with GPG, and you can be on your way with using it in a matter of minutes. In this tutorial, I’ll show you how to encrypt and decrypt files with GPG.***

To install:

sudo apt install gpg

To generate keys:

gpg --full-generate-key

To check the keys:

To check the public keys:

gpg --list-public-keys

To check the private keys:

gpg --list-secret-keys

To encrypt:

gpg --encrypt --output <file name to encrypt e.g. text.txt>.gpg --recipient <receipent's email address> <file name to encrypt>

For example: *gpg --encrypt --output test.txt.gpg --recipient alphausjad12@gmail.com test.txt*

To check the files in the directory:

ls

To display the file contents:

cat <file name>

For example: *cat test.txt.gpg*

To decrypt:

gpg –decrypt –output <output file name> <encrypted file name>

For example: *gpg --decrypt --output decrypted.txt test.txt.gpg*

To export a key:

gpg --output <key name>.gpg --export <key ID>

For example: *gpg --output key.gpg --export 6D5895935478B8EC5AC40D751D5D0D637B956521*

To import a key:

gpg –import <key file name>

For example: *gpg --import key.gpg*

To verify the key:

gpg --edit-key <key ID>

For example: *gpg --edit-key 6D5895935478B8EC5AC40D751D5D0D637B956521*

Once ‘edit’ command is executed:

To check the fingerprint of the key:

enter 'fpr'

To sign the key:

enter 'sign'