A Practical GPG Guide Part 3: Encrypt and Decrypt Files

Xiao Guoan (Admin)

4-5 minutes

In part 2, I explained how to upload a public key to a key server and import public key to a local keyring. In part 3, you will learn how to encrypt a file with public key and decrypt it with private key from the command line.

How GPG Encryption Works

If you need to send an encrypted file to a recipient with GPG, follow these steps

- Import the recipient's public key to your keyring.
- Encrypt the file with the recipient's public key
- Send the encrypted file to the recipient.
- The recipient decrypts the file with his/her own private key.

Step 1: Create a Second User Account

We will need another user account for testing. Run the following command to create the `test` user account, which will act as the file sender.

sudo adduser test

Enter the `sudo` password, then set a password for the `test` user account.

```
linuxbabe@ubuntu:~$ sudo adduser test
[sudo] password for linuxbabe:
Adding user `test' ...
Adding new group `test' (1003) ...
Adding new user `test' (1003) with group `test' ...
Creating home directory `/home/test' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for test
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
linuxbabe@ubuntu:~$
```

Step 2: Import the Public Key

Switch to the `test` user account. (Please don't leave out the dash character.)

su - test

Because we use the `test` account as the file sender, it doesn't need its own GPG key, we just need to import the recipient's public key. In part 2, we uploaded the public key to a key server with the following command:

gpg --send-key key-id

So now you can just run the following command to import the public key. User ID is your GPG email address.

gpg --search user-id

```
linuxbabe@ubuntu:~$ su - test
Password:
test@ubuntu:~$ gpg --search xiao@linuxbabe.com
gpg: data source: https://keys.openpgp.org:443
(1)     Xiao Guoan <xiao@linuxbabe.com>
          256 bit EDDSA key 752E173A3F8B04F5, created: 2022-05-10
Keys 1-1 of 1 for "xiao@linuxbabe.com".  Enter number(s), N)ext, or Q)uit > 1
gpg: key 752E173A3F8B04F5: public key "Xiao Guoan <xiao@linuxbabe.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
test@ubuntu:~$
```

As you can see, it found one record of my email address on the key server, so enter number 1 to import this key. Then check the fingerprint of this key:

gpg --fingerprint user-id

```
test@ubuntu:~$ gpg --fingerprint xiao@linuxbabe.com
pub   ed25519 2022-05-10 [SC] [expires: 2024-05-09]
      378C B32D 8AC7 D656 F389  61B1 752E 173A 3F8B 04F5
uid           [ unknown] Xiao Guoan <xiao@linuxbabe.com>
sub   cv25519 2022-05-10 [E] [expires: 2024-05-09]

test@ubuntu:~$
```

The fingerprint of the imported key is *378C B32D 8AC7 D656 F389 61B1 752E 173A 3F8B 04F5*.

Now open another terminal window, so you will be using the original account, and check the fingerprint of the GPG key.

```
linuxbabe@ubuntu:~$ gpg --fingerprint xiao@linuxbabe.com
pub   ed25519 2022-05-10 [SC] [expires: 2024-05-09]
      378C B32D 8AC7 D656 F389  61B1 752E 173A 3F8B 04F5
uid           [ultimate] Xiao Guoan <xiao@linuxbabe.com>
sub   cv25519 2022-05-10 [E] [expires: 2024-05-09]

linuxbabe@ubuntu:~$
```

As you can see, the two fingerprints match, so it's the correct key.

Hint: When you receive a person's public key, you must contact the person by email, over the phone, or in-person to ask them if it's the correct fingerprint. **If the two fingerprints match, then you get the correct public key.**

In the real world, you should also run the following command to sign the recipient's public key. However we are testing, so you don't need to do it now.

gpg --sign-key key-id

Step 3: Encrypt File With Public Key

Using the `test` account, run the following command to create a sample file.

echo "This file is encrypted with GPG" | tee test-file.txt

Then run the following command to encrypt the file for a single recipient. `--armor` means the file will be ASCII armored instead of creating a binary file.

gpg --recipient user-id --encrypt --armor test-file.txt

Notice the warning "There's no assurance this key belongs to the named user." This is because we didn't sign the recipient's public

key in the previous step. Press y and `Enter`. It will create a file with `.asc` file extension, which is the encrypted file, also known as ciphertext.

If you have imported multiple public keys from multiple people, you can use the following syntax to encrypt a file for multiple recipients.

gpg --recipient user-id1 --recipient user-id2 --encrypt --armor test-file.txt

Step 4: Decrypt File with Private Key

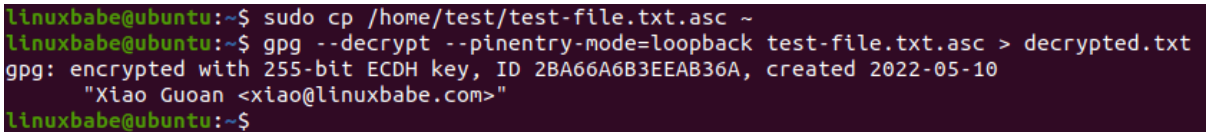
Now switch back to the original account and copy the `test-file.txt.asc` file.

sudo cp /home/test/test-file.txt.asc ~

Then enter the following command to decrypt it.

gpg --decrypt --pinentry-mode=loopback test-file.txt.asc > decrypted.txt

It will ask you to enter the passphrase to unlock your private key. After that, the decrypted content will be saved as `decrypted.txt`.



Now you can check the content of `decrypted.txt`.

cat decrypted.txt

output:

This file is encrypted with GPG

Next Step

Now you learned how to encrypt and decrypt files with GPG from the command line. In part 4, we will learn how to configure GPG in the Thunderbird email client, so you don’t have to type commands.