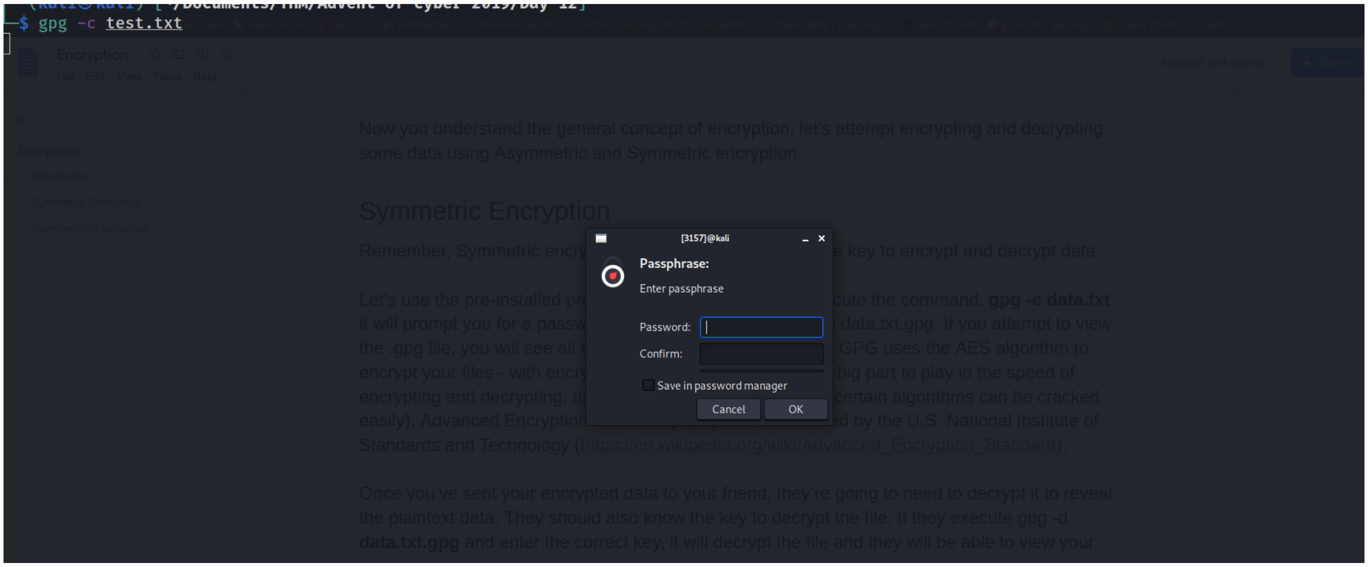
Cryptography

⟹ Symmetric encryption : This type of encryption uses same key to encrypt and decrypt data

Example :

using the gpg (default tool for kali linux) we can encrypt or decrypt any file

```
gpg -c <filename>
```



```
┌──(kali㉿kali)-[~/Documents/THM/Advent of Cyber 2019/Day 12]
└─$ gpg -c test.txt
```

Now you understand the general concept of encryption, let's attempt encrypting and decrypting some data using Asymmetric and Symmetric encryption.

Symmetric Encryption

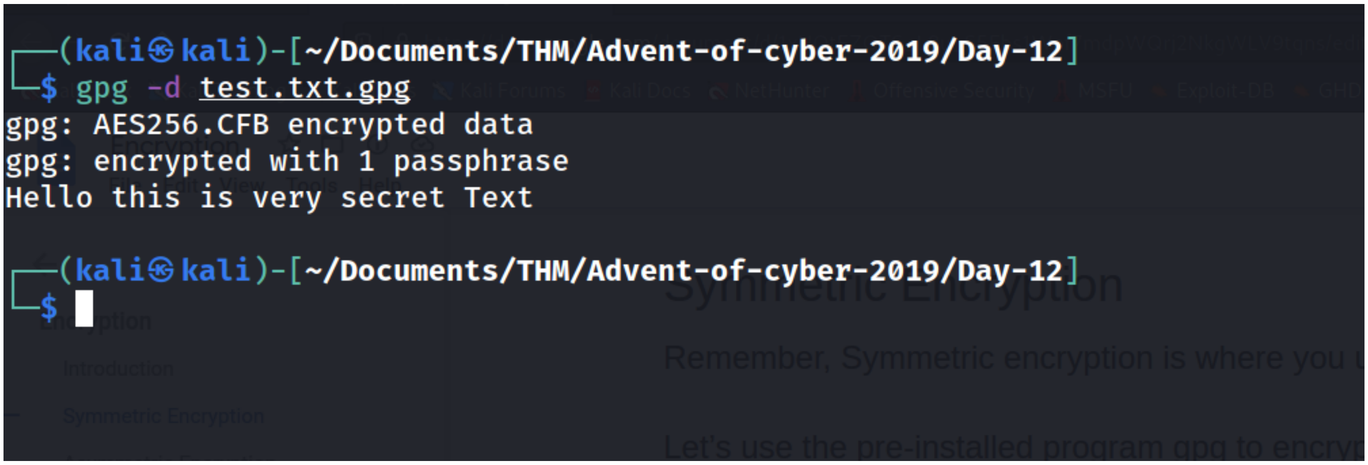
Remember, Symmetric encry… …e key to encrypt and decrypt data.

Let's use the pre-installed pr… …cute the command: **gpg -c data.txt**
it will prompt you for a passw… …data.txt.gpg. If you attempt to view
the .gpg file, you will see all c… …GPG uses the AES algorithm to
encrypt your files - with encry… …big part to play in the speed of
encrypting and decrypting, a… …certain algorithms can be cracked
easily). Advanced Encryption… …ed by the U.S. National Institute of
Standards and Technology (https://en.wikipedia.org/wiki/Advanced_Encryption_Standard).

Once you've sent your encrypted data to your friend, they're going to need to decrypt it to reveal the plaintext data. They should also know the key to decrypt the file. If they execute gpg -d **data.txt.gpg** and enter the correct key, it will decrypt the file and they will be able to view your

⟶ here we enter password to decrypt file and then it will generate .gpg file

```
gpg -d <filename>
```



```
┌──(kali㉿kali)-[~/Documents/THM/Advent-of-cyber-2019/Day-12]
└─$ gpg -d test.txt.gpg
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
Hello this is very secret Text

┌──(kali㉿kali)-[~/Documents/THM/Advent-of-cyber-2019/Day-12]
└─$ 
```

⇒ Asymmetric encryption : This type of encryption uses 2 keys to encrypt and decrypt

Example :

SSH keys uses asymmetric encryption . they use private key and public keys for authentication

you place your public key to the server and you use your private key to login to the ssh

To generate a private key we use the following command (8912 creates the key 8912 bits long):

```
openssl genrsa -aes256 -out private.key 8912
```

To generate a public key we use our previously generated private key:

```
openssl rsa -in private.key -pubout -out public.key
```

Lets now encrypt a file (plaintext.txt) using our public key:

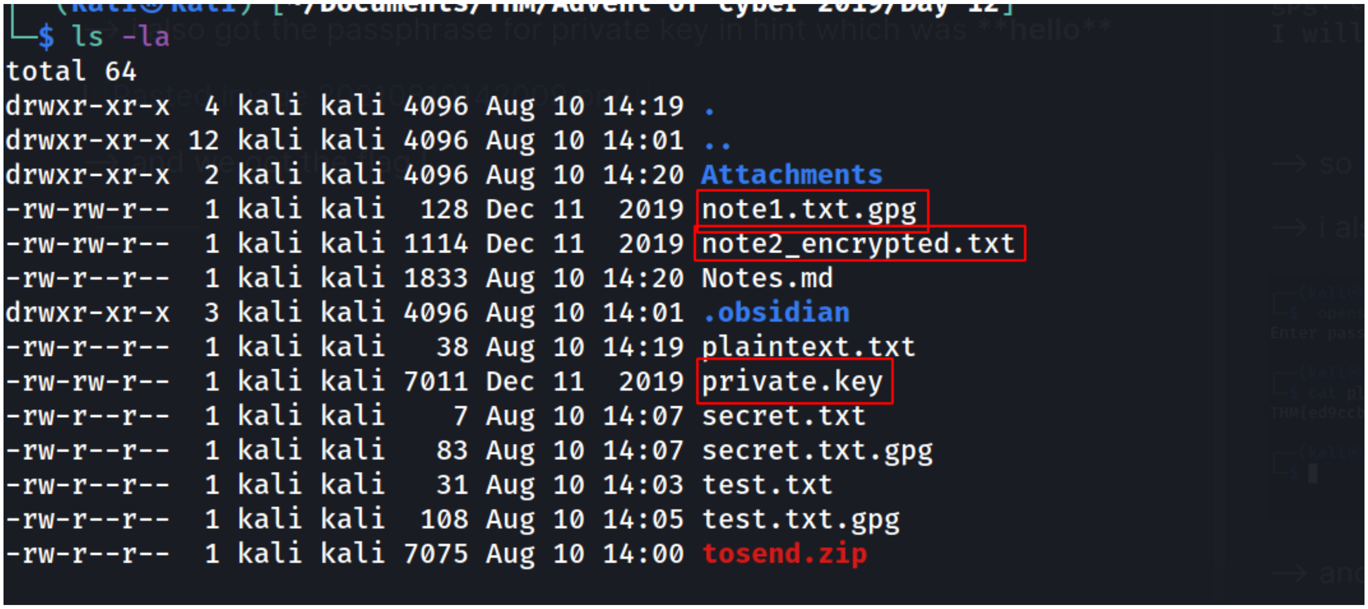
```
openssl rsautl -encrypt -pubin -inkey public.key -in plaintext.txt
```

Now, if we use our private key, we can decrypt the file and get the original message:

```
openssl rsautl -decrypt -inkey private.key -in encrypted.txt -out p
```

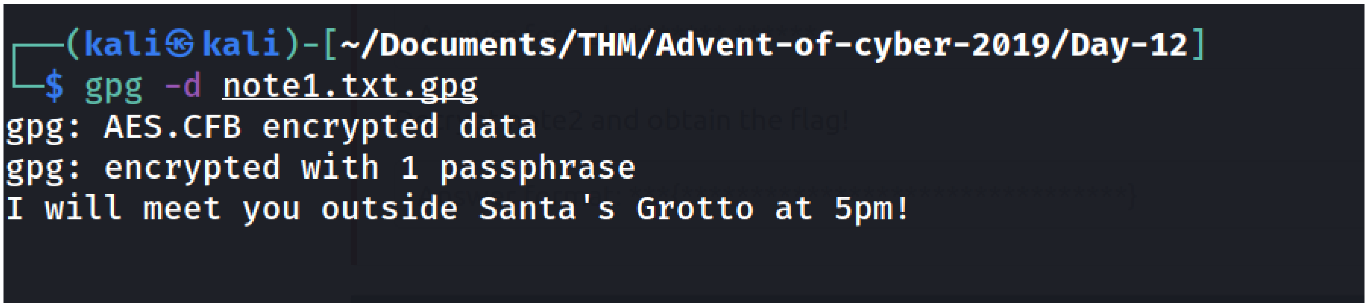
Challenge

→ we have 3 files so we have to decrypt them so Let's start with note1



```
┌──(kali㉿kali)-[~/Documents/THM/Advent-of-cyber-2019/Day-12]
└─$ ls -la
total 64
drwxr-xr-x  4 kali kali 4096 Aug 10 14:19 .
drwxr-xr-x 12 kali kali 4096 Aug 10 14:01 ..
drwxr-xr-x  2 kali kali 4096 Aug 10 14:20 Attachments
-rw-rw-r--  1 kali kali  128 Dec 11  2019 note1.txt.gpg
-rw-rw-r--  1 kali kali 1114 Dec 11  2019 note2_encrypted.txt
-rw-r--r--  1 kali kali 1833 Aug 10 14:20 Notes.md
drwxr-xr-x  3 kali kali 4096 Aug 10 14:01 .obsidian
-rw-r--r--  1 kali kali   38 Aug 10 14:19 plaintext.txt
-rw-rw-r--  1 kali kali 7011 Dec 11  2019 private.key
-rw-r--r--  1 kali kali    7 Aug 10 14:07 secret.txt
-rw-r--r--  1 kali kali   83 Aug 10 14:07 secret.txt.gpg
-rw-r--r--  1 kali kali   31 Aug 10 14:03 test.txt
-rw-r--r--  1 kali kali  108 Aug 10 14:05 test.txt.gpg
-rw-r--r--  1 kali kali 7075 Aug 10 14:00 tosend.zip
```

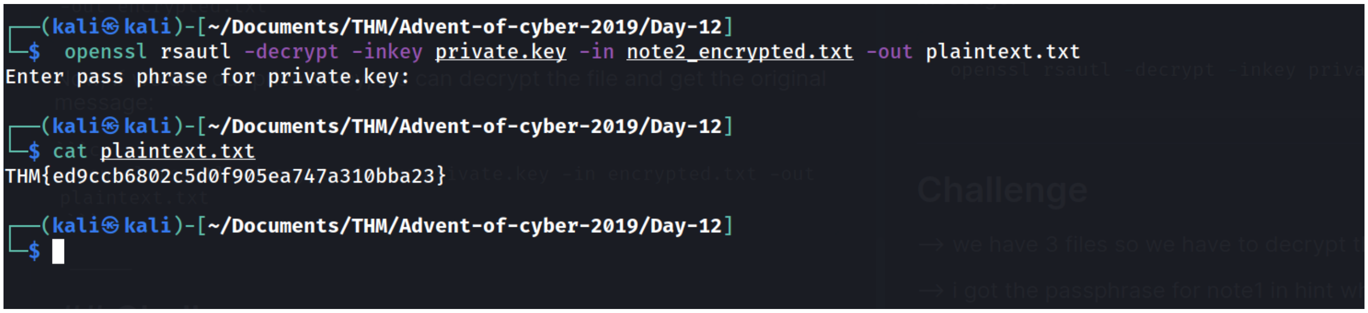
→ i got the passphrase for note1 in hint which was **25daysofchristmas**



```
┌──(kali㉿kali)-[~/Documents/THM/Advent-of-cyber-2019/Day-12]
└─$ gpg -d note1.txt.gpg
gpg: AES.CFB encrypted data
gpg: encrypted with 1 passphrase
I will meet you outside Santa's Grotto at 5pm!
```

→ so Let's decrypt second file using private.key

→ i also got the passphrase for private key in hint which was **hello**



```
┌──(kali㉿kali)-[~/Documents/THM/Advent-of-cyber-2019/Day-12]
└─$  openssl rsautl -decrypt -inkey private.key -in note2_encrypted.txt -out plaintext.txt
Enter pass phrase for private.key:

┌──(kali㉿kali)-[~/Documents/THM/Advent-of-cyber-2019/Day-12]
└─$ cat plaintext.txt
THM{ed9ccb6802c5d0f905ea747a310bba23}

┌──(kali㉿kali)-[~/Documents/THM/Advent-of-cyber-2019/Day-12]
└─$ █
```

→ and we got the flag !