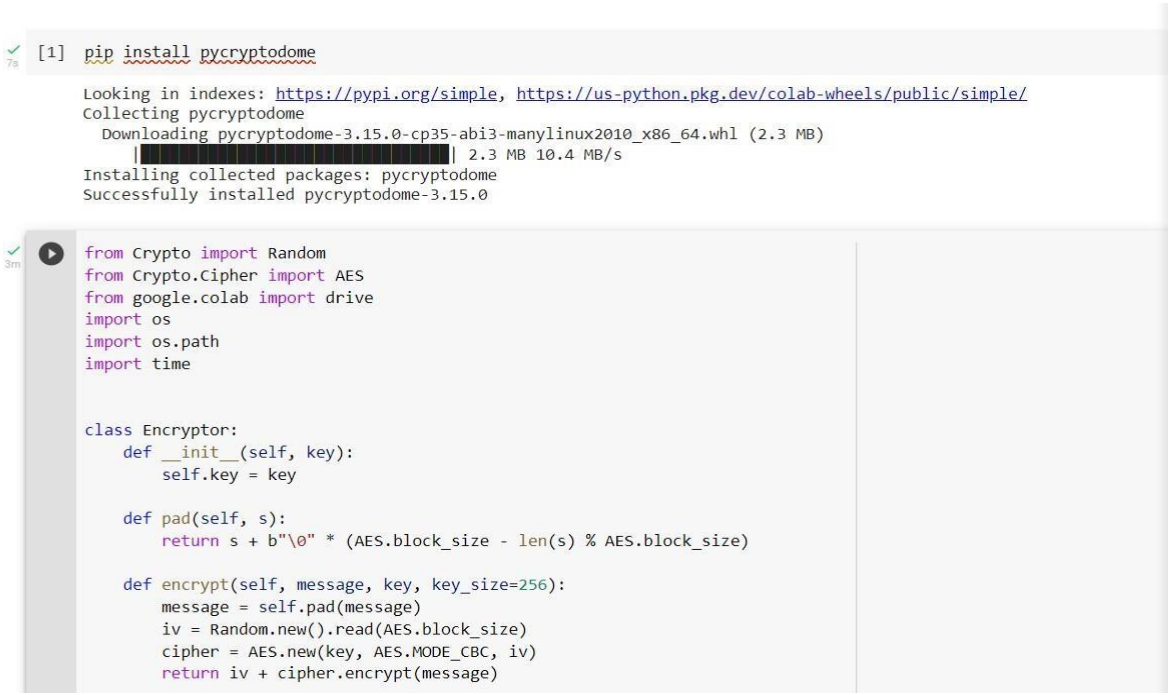
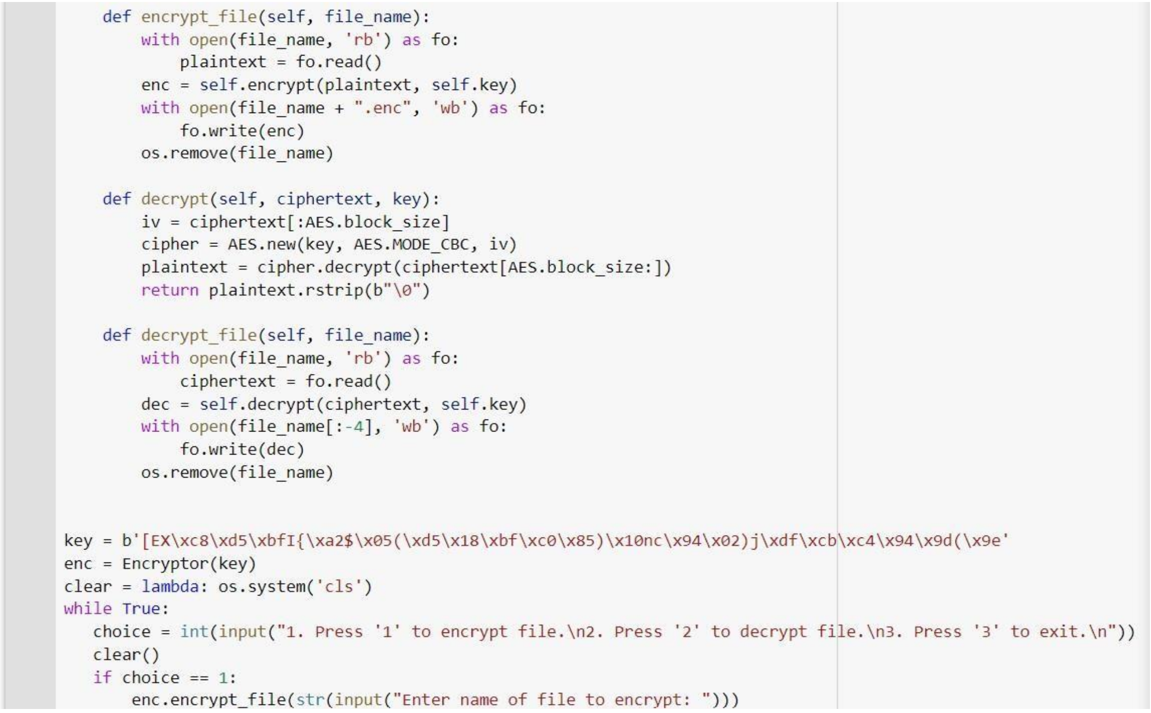


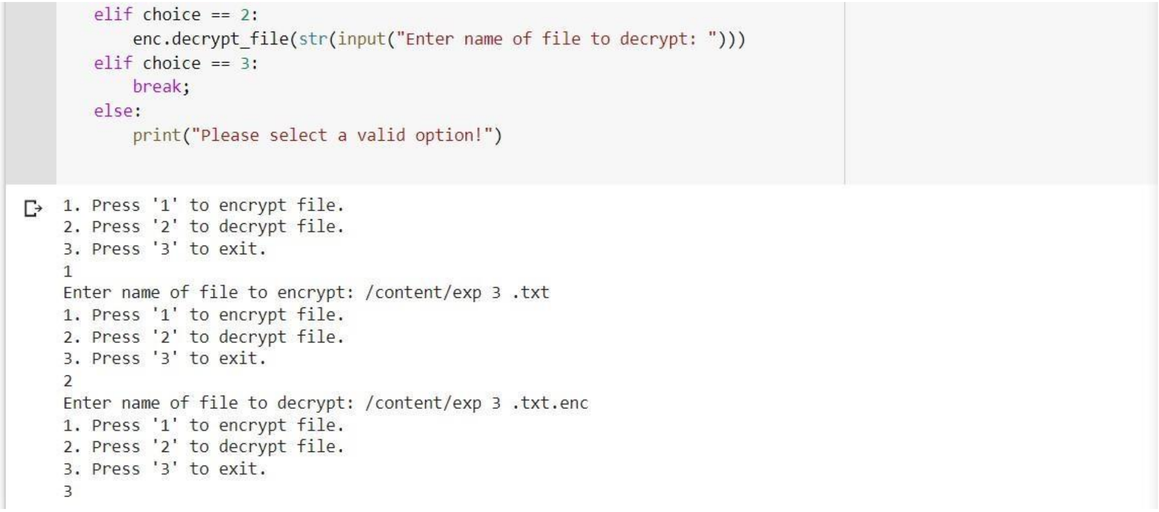
```
[1] pip install pycryptodome
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pycryptodome
  Downloading pycryptodome-3.15.0-cp35-abi3-manylinux2010_x86_64.whl (2.3 MB)
     |████████████████████████████████| 2.3 MB 10.4 MB/s
Installing collected packages: pycryptodome
Successfully installed pycryptodome-3.15.0
```

```
from Crypto import Random
from Crypto.Cipher import AES
from google.colab import drive
import os
import os.path
import time


class Encryptor:
    def __init__(self, key):
        self.key = key

    def pad(self, s):
        return s + b"\0" * (AES.block_size - len(s) % AES.block_size)

    def encrypt(self, message, key, key_size=256):
        message = self.pad(message)
        iv = Random.new().read(AES.block_size)
        cipher = AES.new(key, AES.MODE_CBC, iv)
        return iv + cipher.encrypt(message)
```

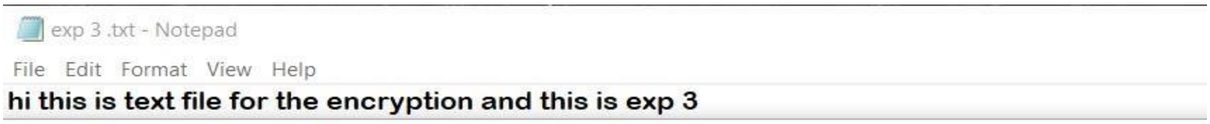


```
    def encrypt_file(self, file_name):
        with open(file_name, 'rb') as fo:
            plaintext = fo.read()
        enc = self.encrypt(plaintext, self.key)
        with open(file_name + ".enc", 'wb') as fo:
            fo.write(enc)
        os.remove(file_name)

    def decrypt(self, ciphertext, key):
        iv = ciphertext[:AES.block_size]
        cipher = AES.new(key, AES.MODE_CBC, iv)
        plaintext = cipher.decrypt(ciphertext[AES.block_size:])
        return plaintext.rstrip(b"\0")

    def decrypt_file(self, file_name):
        with open(file_name, 'rb') as fo:
            ciphertext = fo.read()
        dec = self.decrypt(ciphertext, self.key)
        with open(file_name[:-4], 'wb') as fo:
            fo.write(dec)
        os.remove(file_name)


key = b'[EX\xc8\xd5\xbfI{\xa2$\x05(\xd5\x18\xbf\xc0\x85)\x10nc\x94\x02)j\xdf\xcb\xc4\x94\x9d(\x9e'
enc = Encryptor(key)
clear = lambda: os.system('cls')
while True:
   choice = int(input("1. Press '1' to encrypt file.\n2. Press '2' to decrypt file.\n3. Press '3' to exit.\n"))
   clear()
   if choice == 1:
       enc.encrypt_file(str(input("Enter name of file to encrypt: ")))
```

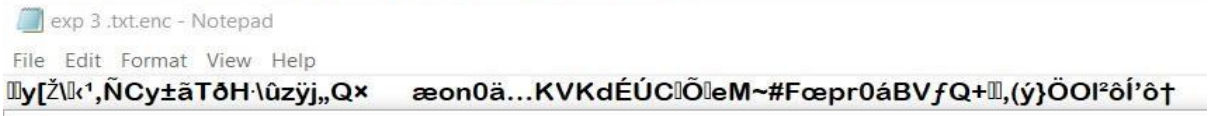


```
   elif choice == 2:
       enc.decrypt_file(str(input("Enter name of file to decrypt: ")))
   elif choice == 3:
       break;
   else:
       print("Please select a valid option!")
```

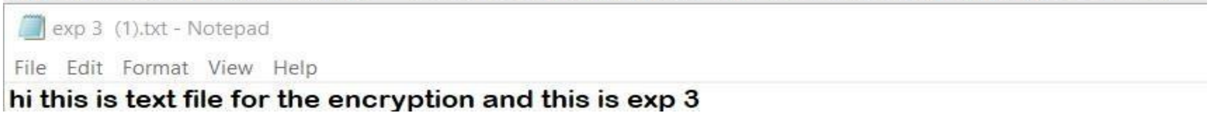
```
1. Press '1' to encrypt file.
2. Press '2' to decrypt file.
3. Press '3' to exit.
1
Enter name of file to encrypt: /content/exp 3 .txt
1. Press '1' to encrypt file.
2. Press '2' to decrypt file.
3. Press '3' to exit.
2
Enter name of file to decrypt: /content/exp 3 .txt.enc
1. Press '1' to encrypt file.
2. Press '2' to decrypt file.
3. Press '3' to exit.
3
```



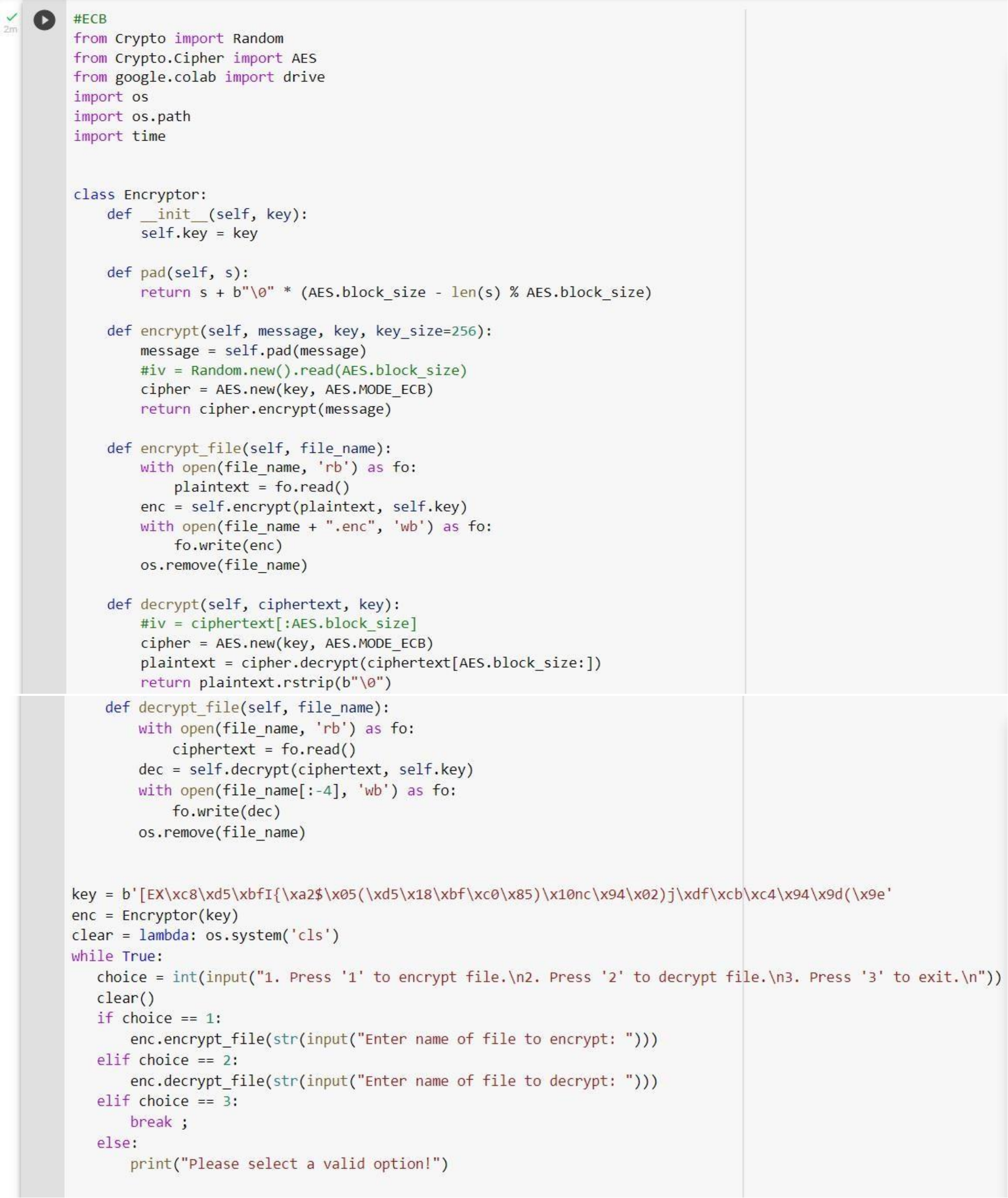
hi this is text file for the encryption and this is exp 3



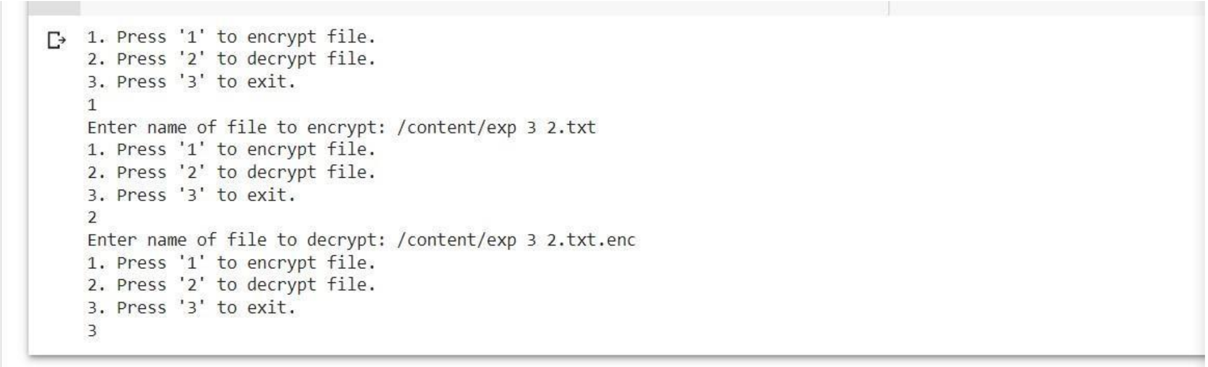
▯y[Ž\▯‹¹,ÑCy±ãTðH·\ûzÿj„Q×   æon0ä…KVKdÉÚC▯Õ▯eM~#Fœpr0áBVƒQ+▯,(ý}ÖOl²ôÍ’ô†



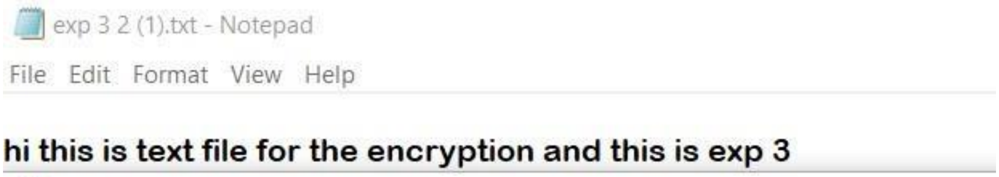
hi this is text file for the encryption and this is exp 3



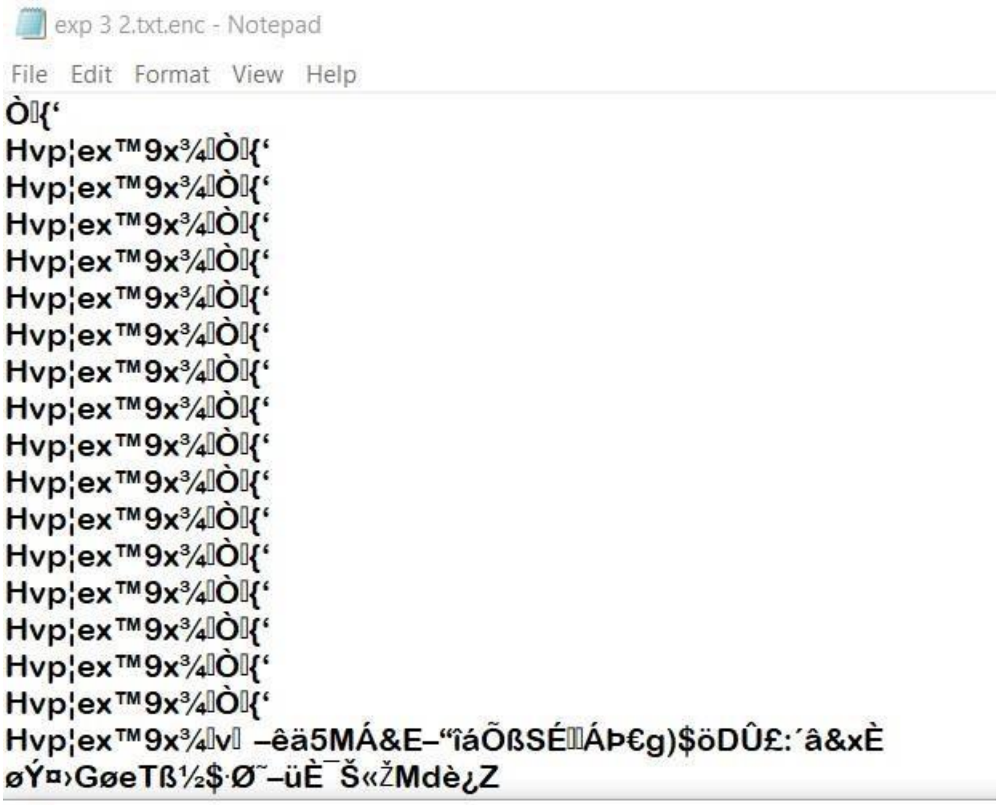
```
#ECB
from Crypto import Random
from Crypto.Cipher import AES
from google.colab import drive
import os
import os.path
import time


class Encryptor:
    def __init__(self, key):
        self.key = key

    def pad(self, s):
        return s + b"\0" * (AES.block_size - len(s) % AES.block_size)

    def encrypt(self, message, key, key_size=256):
        message = self.pad(message)
        #iv = Random.new().read(AES.block_size)
        cipher = AES.new(key, AES.MODE_ECB)
        return cipher.encrypt(message)

    def encrypt_file(self, file_name):
        with open(file_name, 'rb') as fo:
            plaintext = fo.read()
        enc = self.encrypt(plaintext, self.key)
        with open(file_name + ".enc", 'wb') as fo:
            fo.write(enc)
        os.remove(file_name)

    def decrypt(self, ciphertext, key):
        #iv = ciphertext[:AES.block_size]
        cipher = AES.new(key, AES.MODE_ECB)
        plaintext = cipher.decrypt(ciphertext[AES.block_size:])
        return plaintext.rstrip(b"\0")
    def decrypt_file(self, file_name):
        with open(file_name, 'rb') as fo:
            ciphertext = fo.read()
        dec = self.decrypt(ciphertext, self.key)
        with open(file_name[:-4], 'wb') as fo:
            fo.write(dec)
        os.remove(file_name)


key = b'[EX\xc8\xd5\xbfI{\xa2$\x05(\xd5\x18\xbf\xc0\x85)\x10nc\x94\x02)j\xdf\xcb\xc4\x94\x9d(\x9e'
enc = Encryptor(key)
clear = lambda: os.system('cls')
while True:
   choice = int(input("1. Press '1' to encrypt file.\n2. Press '2' to decrypt file.\n3. Press '3' to exit.\n"))
   clear()
   if choice == 1:
       enc.encrypt_file(str(input("Enter name of file to encrypt: ")))
   elif choice == 2:
       enc.decrypt_file(str(input("Enter name of file to decrypt: ")))
   elif choice == 3:
       break ;
   else:
       print("Please select a valid option!")
```



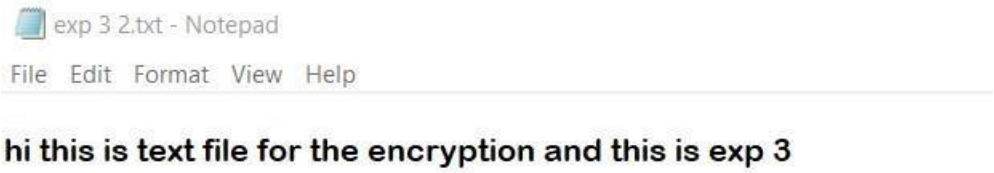
```
1. Press '1' to encrypt file.
2. Press '2' to decrypt file.
3. Press '3' to exit.
1
Enter name of file to encrypt: /content/exp 3 2.txt
1. Press '1' to encrypt file.
2. Press '2' to decrypt file.
3. Press '3' to exit.
2
Enter name of file to decrypt: /content/exp 3 2.txt.enc
1. Press '1' to encrypt file.
2. Press '2' to decrypt file.
3. Press '3' to exit.
3
```



hi this is text file for the encryption and this is exp 3

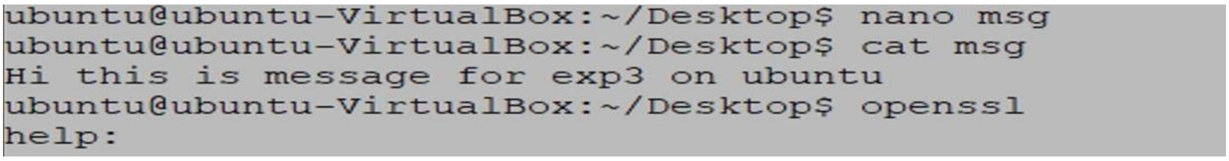


```
Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□Ò□{‘
Hvp¦ex™9x³⁄4□v□ –êä5MÁ&E–“îáÕßSÉ□□ÁÞ€g)$öDÛ£:´â&xÈ
øÝ¤›GøeTß½$·Ø˜–üÈ¯Š«ŽMdè¿Z
```

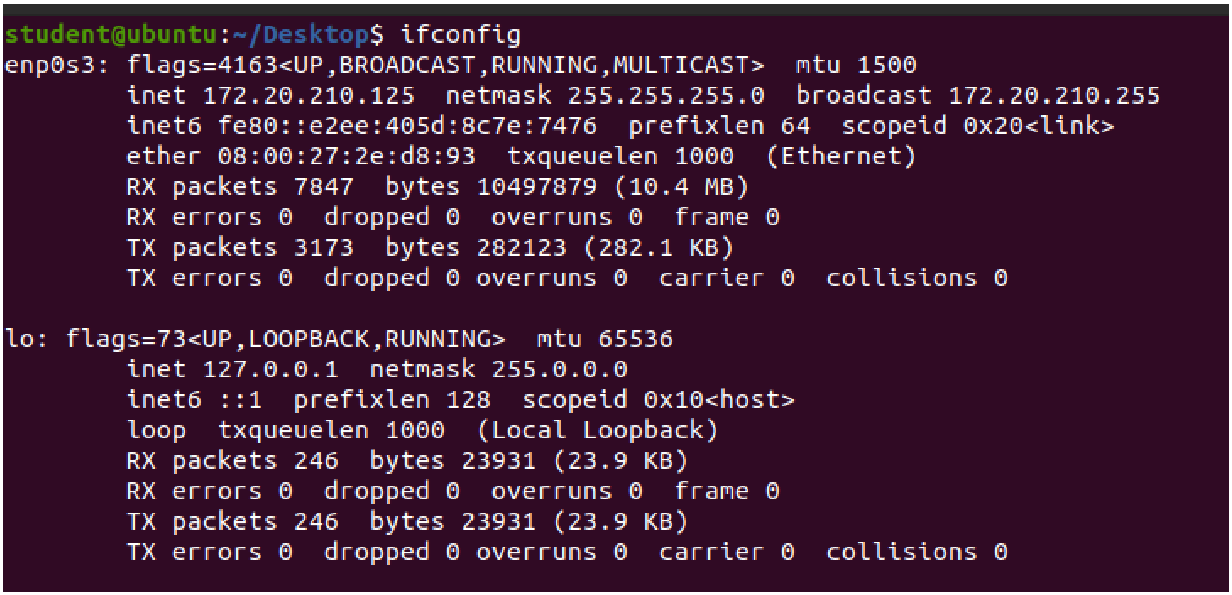


hi this is text file for the encryption and this is exp 3

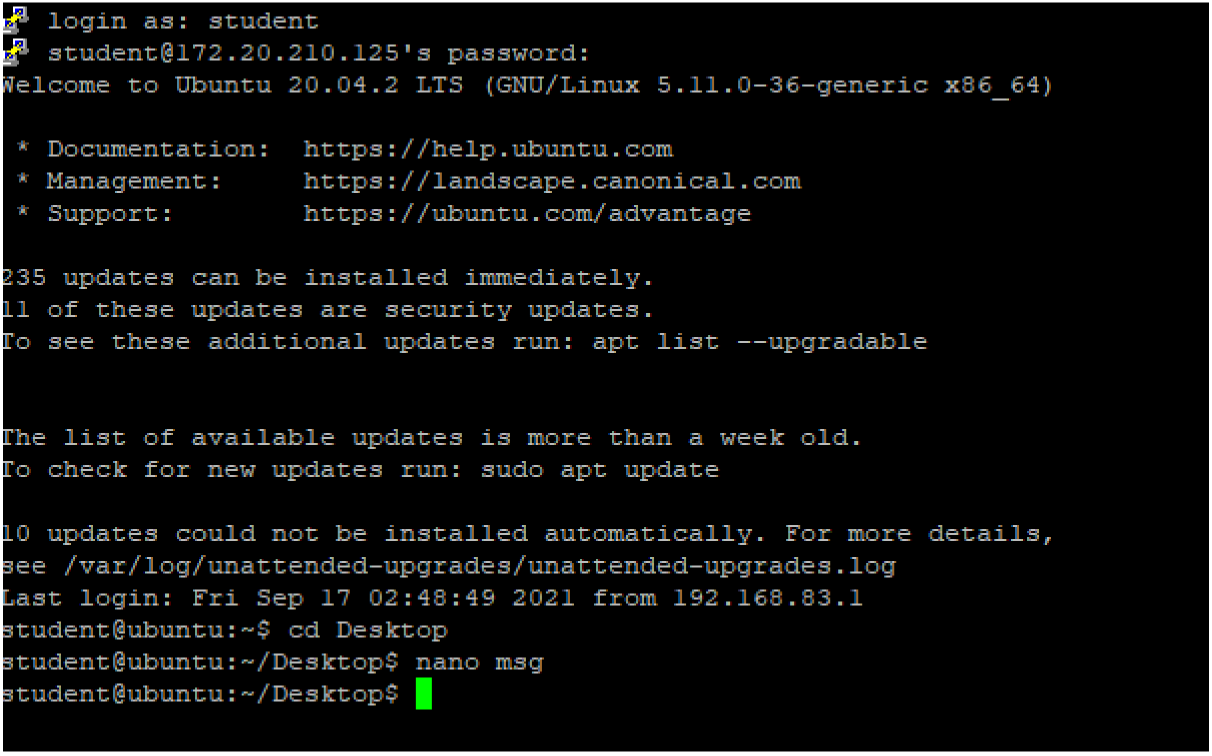
In Ubuntu (linux)



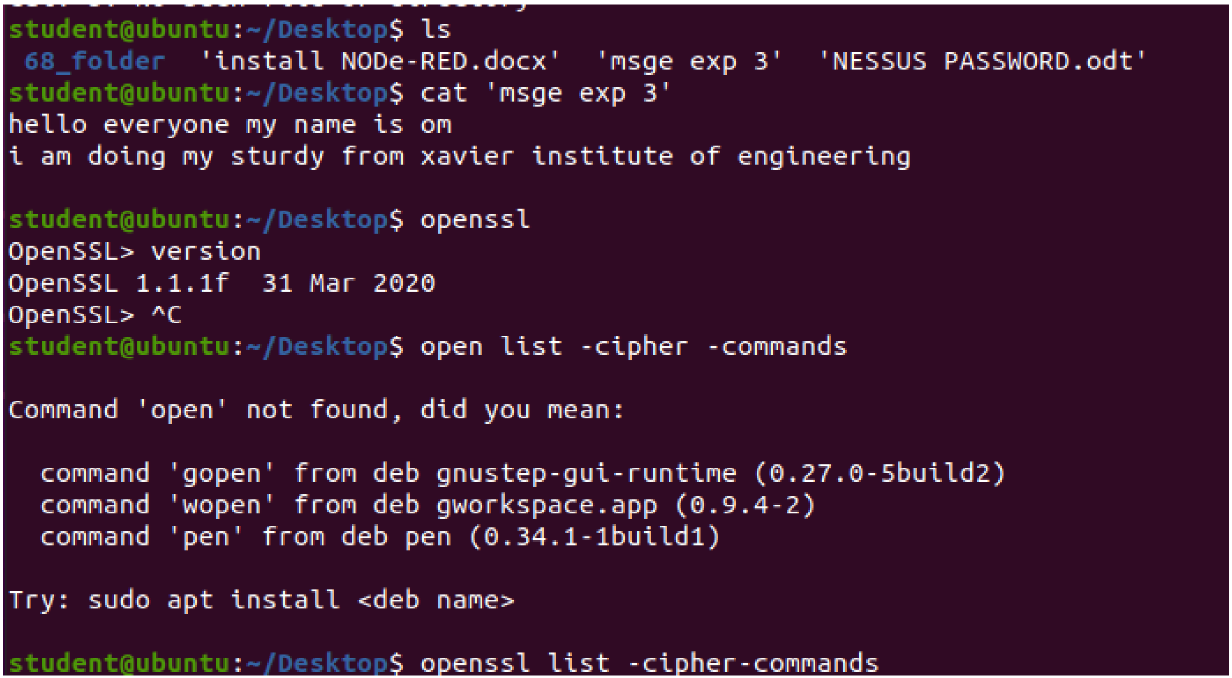
```
ubuntu@ubuntu-VirtualBox:~/Desktop$ nano msg
ubuntu@ubuntu-VirtualBox:~/Desktop$ cat msg
Hi this is message for exp3 on ubuntu
ubuntu@ubuntu-VirtualBox:~/Desktop$ openssl
help:
```



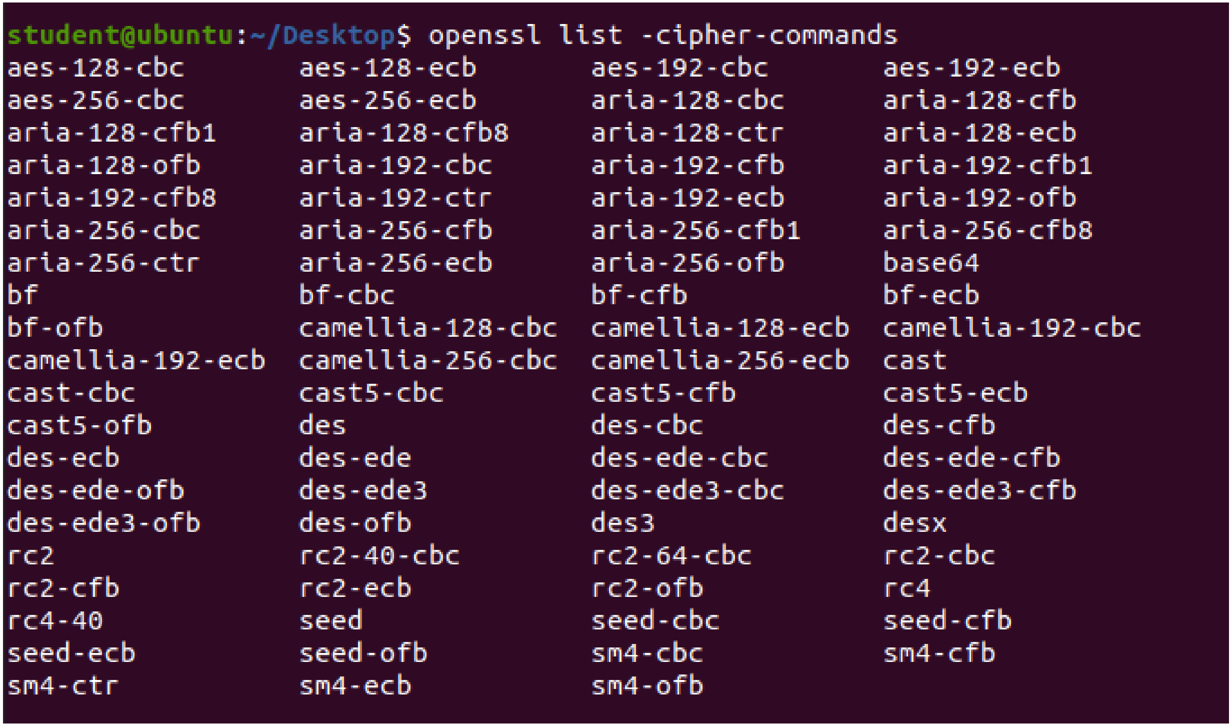
```
student@ubuntu:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.20.210.125  netmask 255.255.255.0  broadcast 172.20.210.255
        inet6 fe80::e2ee:405d:8c7e:7476  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:2e:d8:93  txqueuelen 1000  (Ethernet)
        RX packets 7847  bytes 10497879 (10.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3173  bytes 282123 (282.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 246  bytes 23931 (23.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 246  bytes 23931 (23.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



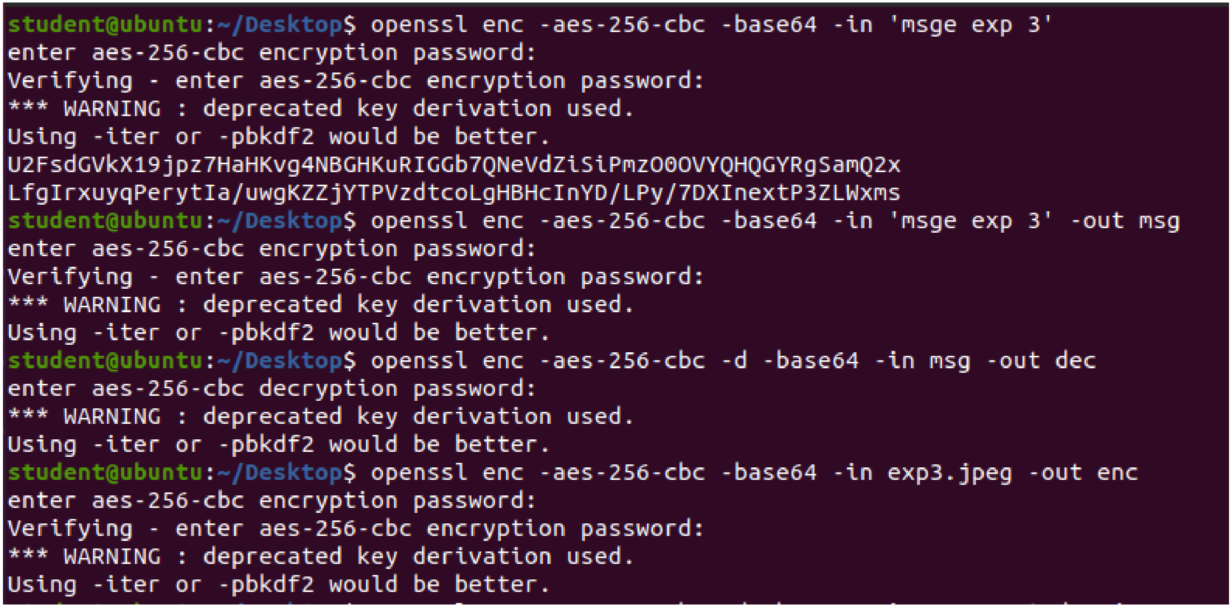
```
  login as: student
  student@172.20.210.125's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.11.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

235 updates can be installed immediately.
11 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

10 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
Last login: Fri Sep 17 02:48:49 2021 from 192.168.83.1
student@ubuntu:~$ cd Desktop
student@ubuntu:~/Desktop$ nano msg
student@ubuntu:~/Desktop$
```



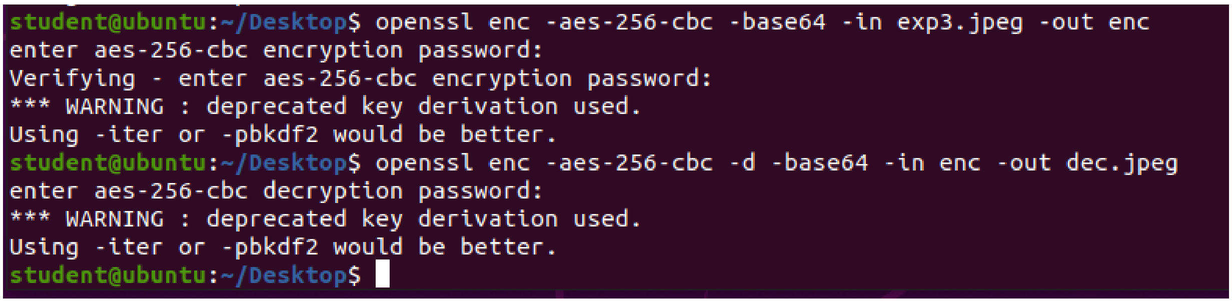
```
student@ubuntu:~/Desktop$ ls
 68_folder   'install NODe-RED.docx'   'msge exp 3'   'NESSUS PASSWORD.odt'
student@ubuntu:~/Desktop$ cat 'msge exp 3'
hello everyone my name is om
i am doing my sturdy from xavier institute of engineering

student@ubuntu:~/Desktop$ openssl
OpenSSL> version
OpenSSL 1.1.1f  31 Mar 2020
OpenSSL> ^C
student@ubuntu:~/Desktop$ open list -cipher -commands

Command 'open' not found, did you mean:

  command 'gopen' from deb gnustep-gui-runtime (0.27.0-5build2)
  command 'wopen' from deb gworkspace.app (0.9.4-2)
  command 'pen' from deb pen (0.34.1-1build1)

Try: sudo apt install <deb name>

student@ubuntu:~/Desktop$ openssl list -cipher-commands
```



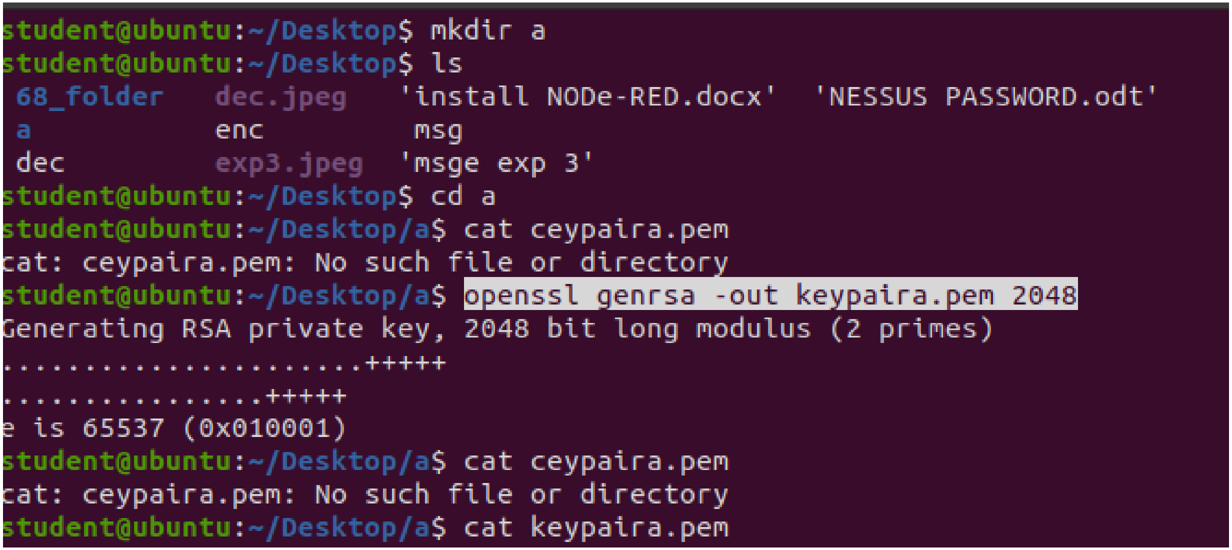
```
student@ubuntu:~/Desktop$ openssl list -cipher-commands
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       aria-128-cbc      aria-128-cfb
aria-128-cfb1     aria-128-cfb8     aria-128-ctr      aria-128-ecb
aria-128-ofb      aria-192-cbc      aria-192-cfb      aria-192-cfb1
aria-192-cfb8     aria-192-ctr      aria-192-ecb      aria-192-ofb
aria-256-cbc      aria-256-cfb      aria-256-cfb1     aria-256-cfb8
aria-256-ctr      aria-256-ecb      aria-256-ofb      base64
bf                bf-cbc            bf-cfb            bf-ecb
bf-ofb            camellia-128-cbc  camellia-128-ecb  camellia-192-cbc
camellia-192-ecb  camellia-256-cbc  camellia-256-ecb  cast
cast-cbc          cast5-cbc         cast5-cfb         cast5-ecb
cast5-ofb         des               des-cbc           des-cfb
des-ecb           des-ede           des-ede-cbc       des-ede-cfb
des-ede-ofb       des-ede3          des-ede3-cbc      des-ede3-cfb
des-ede3-ofb      des-ofb           des3              desx
rc2               rc2-40-cbc        rc2-64-cbc        rc2-cbc
rc2-cfb           rc2-ecb           rc2-ofb           rc4
rc4-40            seed              seed-cbc          seed-cfb
seed-ecb          seed-ofb          sm4-cbc           sm4-cfb
sm4-ctr           sm4-ecb           sm4-ofb
```



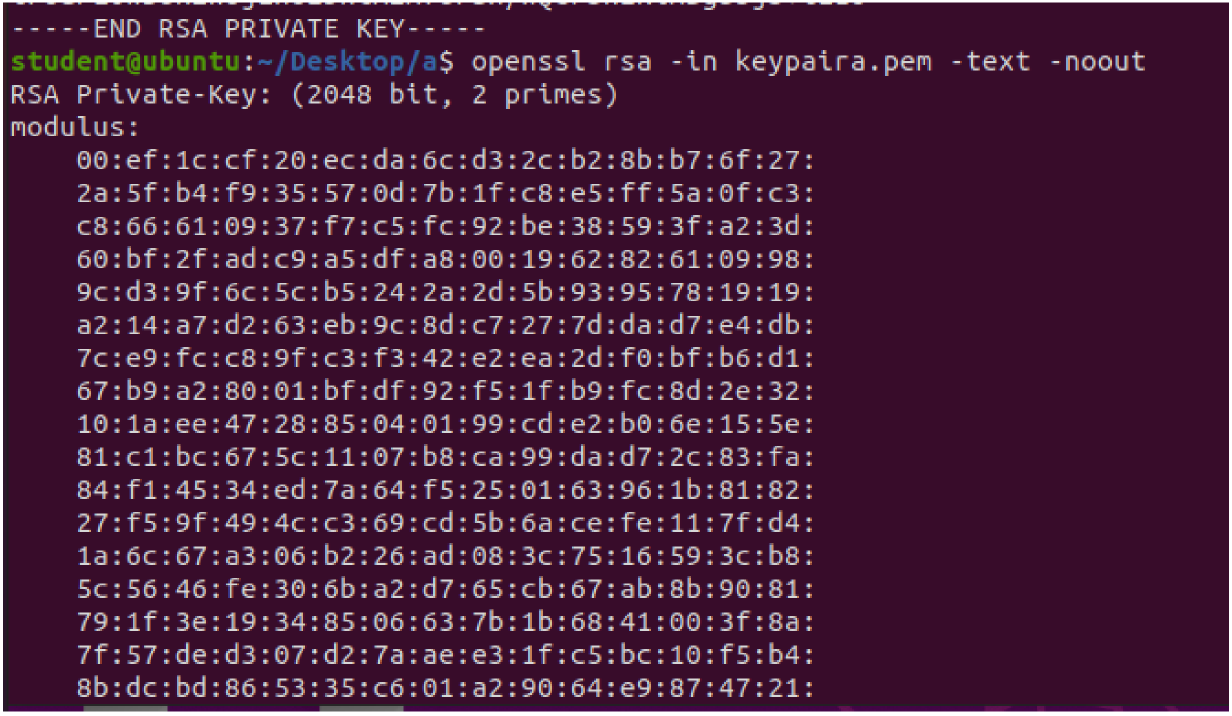
```
student@ubuntu:~/Desktop$ openssl enc -aes-256-cbc -base64 -in 'msge exp 3'
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
U2FsdGVkX19jpz7HaHKvg4NBGHKuRIGGb7QNeVdZiSiPmzO0OVYQHQGYRgSamQ2x
LfgIrxuyqPerytIa/uwgKZZjYTPVzdtcoLgHBHcInYD/LPy/7DXInextP3ZLWxms
student@ubuntu:~/Desktop$ openssl enc -aes-256-cbc -base64 -in 'msge exp 3' -out msg
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
student@ubuntu:~/Desktop$ openssl enc -aes-256-cbc -d -base64 -in msg -out dec
enter aes-256-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
student@ubuntu:~/Desktop$ openssl enc -aes-256-cbc -base64 -in exp3.jpeg -out enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```



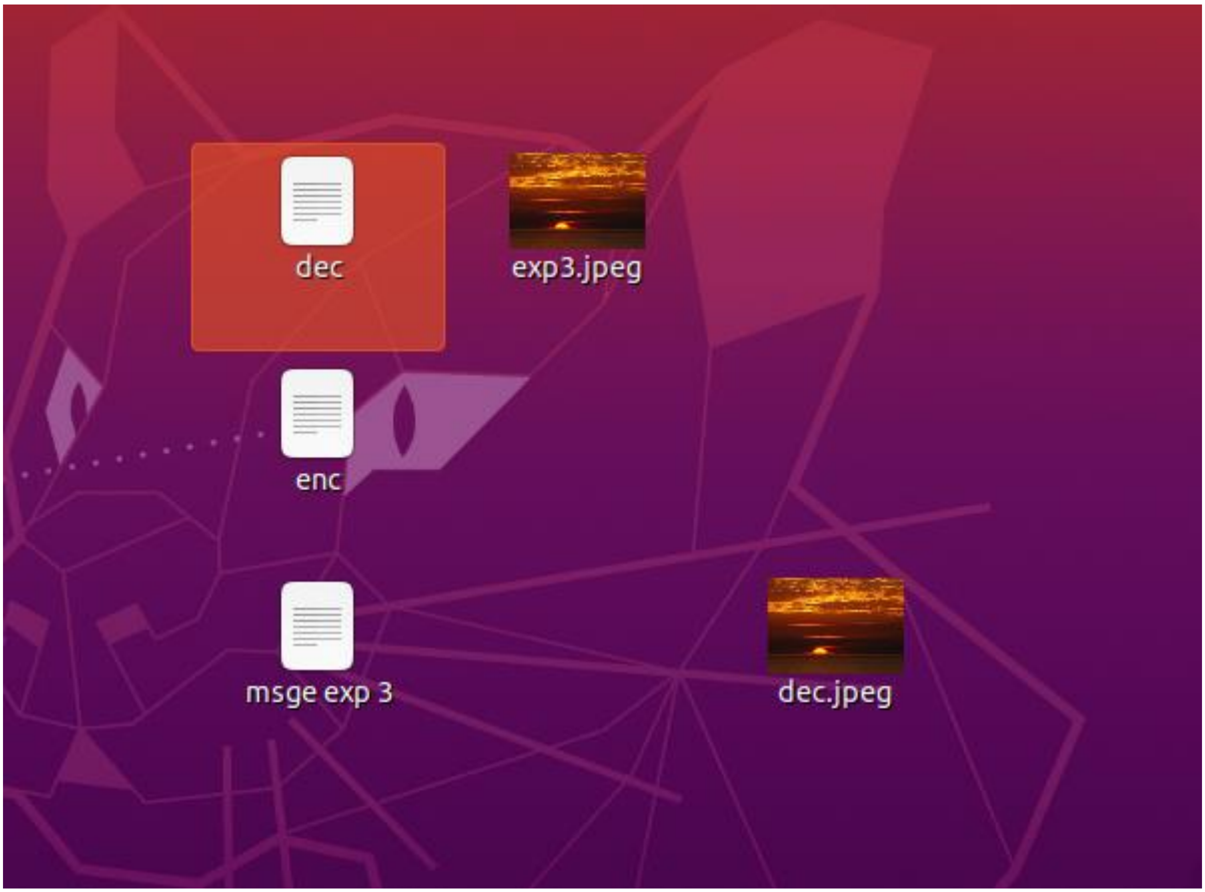
```
student@ubuntu:~/Desktop$ openssl enc -aes-256-cbc -base64 -in exp3.jpeg -out enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
student@ubuntu:~/Desktop$ openssl enc -aes-256-cbc -d -base64 -in enc -out dec.jpeg
enter aes-256-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
student@ubuntu:~/Desktop$ 
```

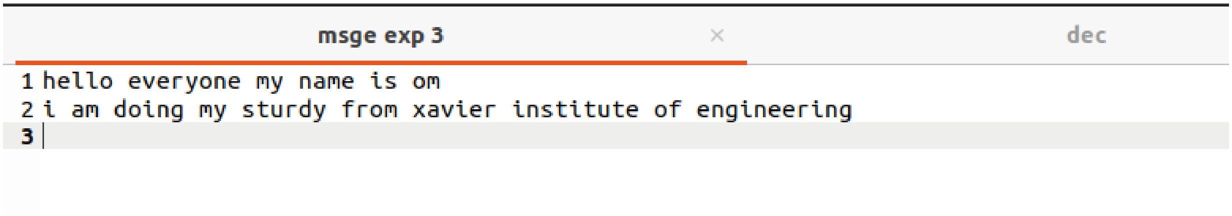


```
student@ubuntu:~/Desktop$ mkdir a
student@ubuntu:~/Desktop$ ls
 68_folder    dec.jpeg    'install NODe-RED.docx'   'NESSUS PASSWORD.odt'
 a            enc          msg
 dec          exp3.jpeg   'msge exp 3'
student@ubuntu:~/Desktop$ cd a
student@ubuntu:~/Desktop/a$ cat ceypaira.pem
cat: ceypaira.pem: No such file or directory
student@ubuntu:~/Desktop/a$ openssl genrsa -out keypaira.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
......................+++++
.................+++++
e is 65537 (0x010001)
student@ubuntu:~/Desktop/a$ cat ceypaira.pem
cat: ceypaira.pem: No such file or directory
student@ubuntu:~/Desktop/a$ cat keypaira.pem
```



```
-----END RSA PRIVATE KEY-----
student@ubuntu:~/Desktop/a$ openssl rsa -in keypaira.pem -text -noout
RSA Private-Key: (2048 bit, 2 primes)
modulus:
    00:ef:1c:cf:20:ec:da:6c:d3:2c:b2:8b:b7:6f:27:
    2a:5f:b4:f9:35:57:0d:7b:1f:c8:e5:ff:5a:0f:c3:
    c8:66:61:09:37:f7:c5:fc:92:be:38:59:3f:a2:3d:
    60:bf:2f:ad:c9:a5:df:a8:00:19:62:82:61:09:98:
    9c:d3:9f:6c:5c:b5:24:2a:2d:5b:93:95:78:19:19:
    a2:14:a7:d2:63:eb:9c:8d:c7:27:7d:da:d7:e4:db:
    7c:e9:fc:c8:9f:c3:f3:42:e2:ea:2d:f0:bf:b6:d1:
    67:b9:a2:80:01:bf:df:92:f5:1f:b9:fc:8d:2e:32:
    10:1a:ee:47:28:85:04:01:99:cd:e2:b0:6e:15:5e:
    81:c1:bc:67:5c:11:07:b8:ca:99:da:d7:2c:83:fa:
    84:f1:45:34:ed:7a:64:f5:25:01:63:96:1b:81:82:
    27:f5:9f:49:4c:c3:69:cd:5b:6a:ce:fe:11:7f:d4:
    1a:6c:67:a3:06:b2:26:ad:08:3c:75:16:59:3c:b8:
    5c:56:46:fe:30:6b:a2:d7:65:cb:67:ab:8b:90:81:
    79:1f:3e:19:34:85:06:63:7b:1b:68:41:00:3f:8a:
    7f:57:de:d3:07:d2:7a:ae:e3:1f:c5:bc:10:f5:b4:
    8b:dc:bd:86:53:35:c6:01:a2:90:64:e9:87:47:21:
```





```
hello everyone my name is om
i am doing my sturdy from xavier institute of engineering
```