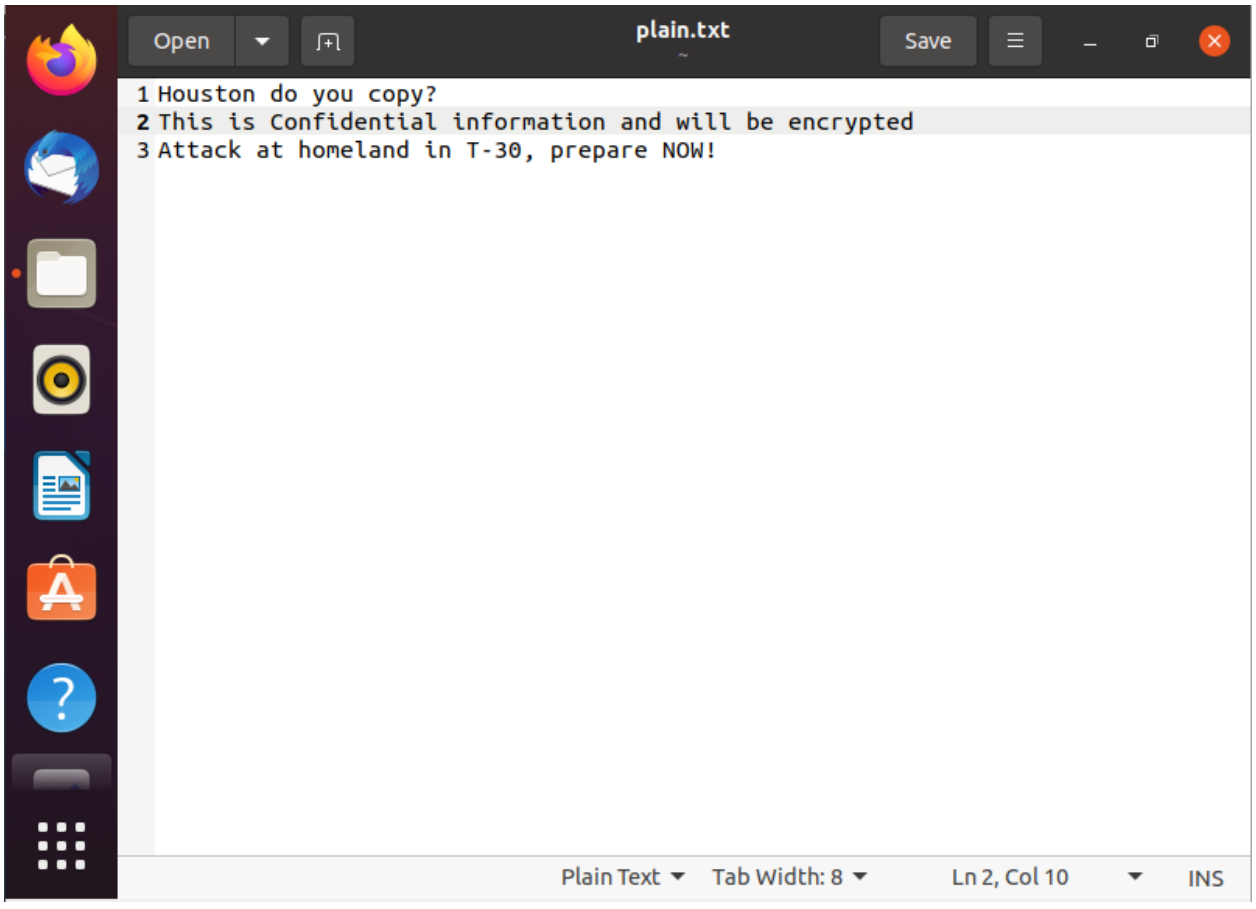
Aditya Motwani
2019130040
TE Comps Batch B

CSS Experiment 3

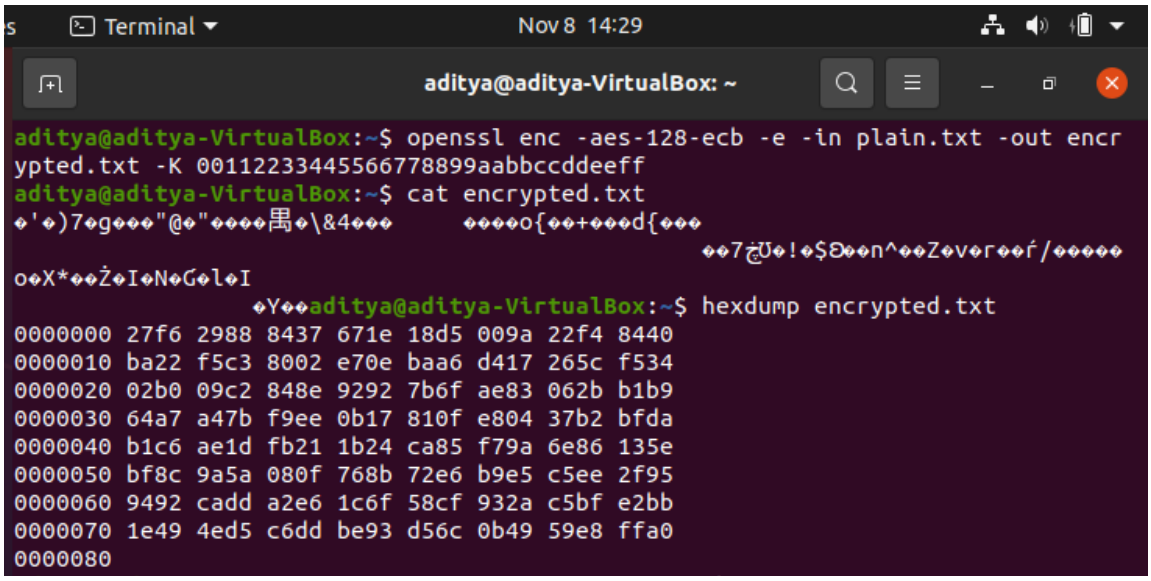
Aim: To get familiar with the concepts in secret-key encryption. To gain first-hand experience on encryption algorithms, encryption modes, paddings, and initial vector (IV) by using tools and writing programs to encrypt/decrypt messages.

Task 1:

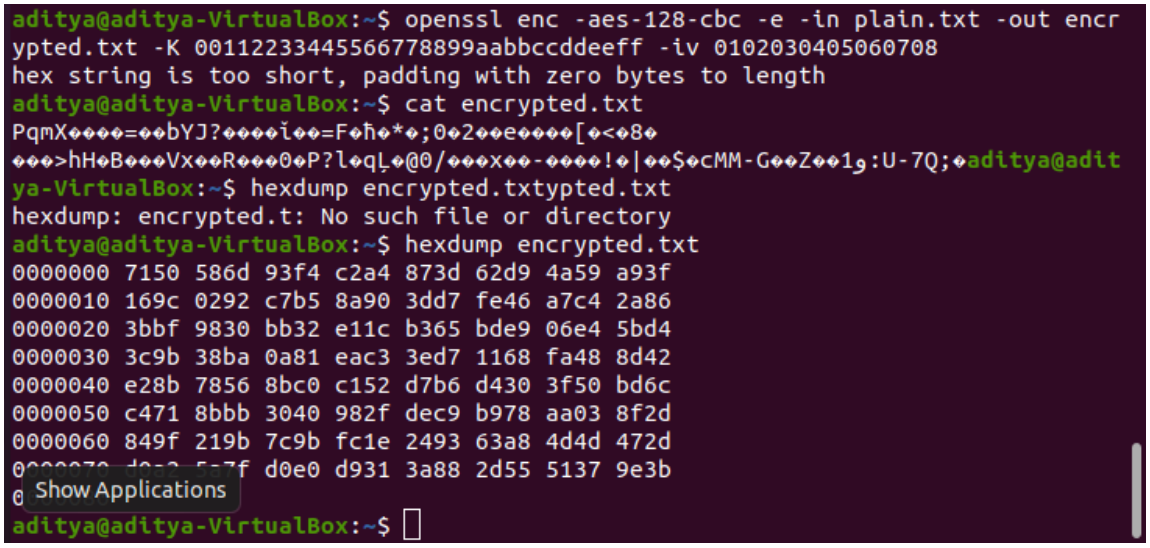
Plain Text:



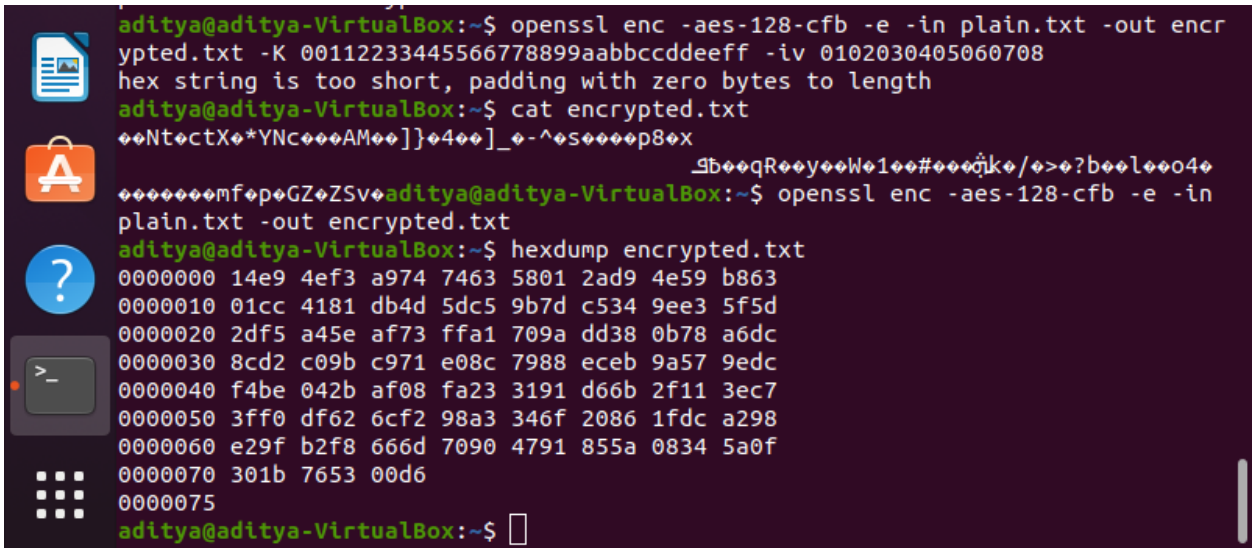
1) Encryption using aes-128-ecb



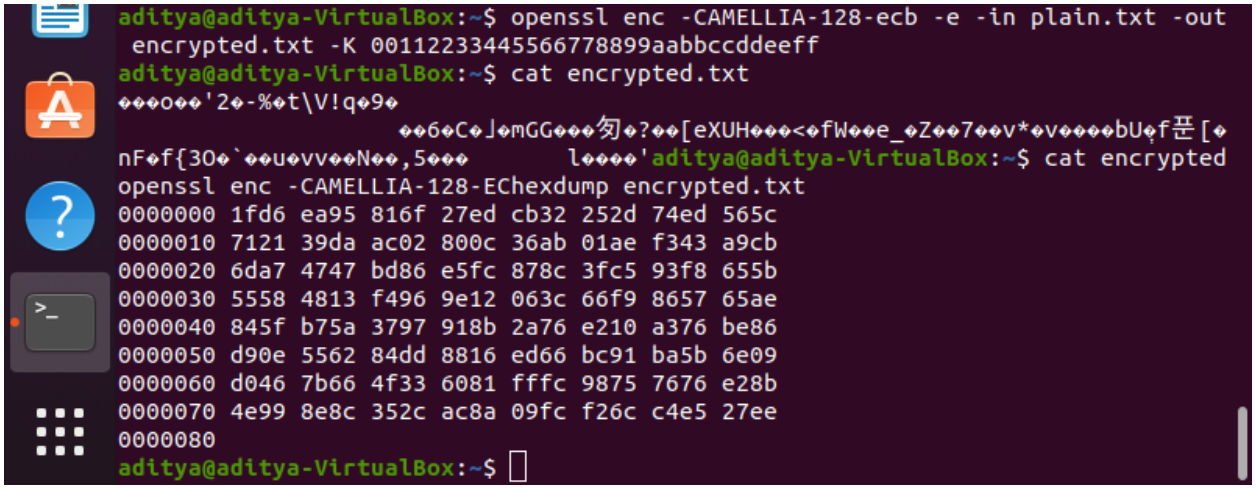
2) Encryption using aes-128-cbc



3) Encryption using aes-128-cfb

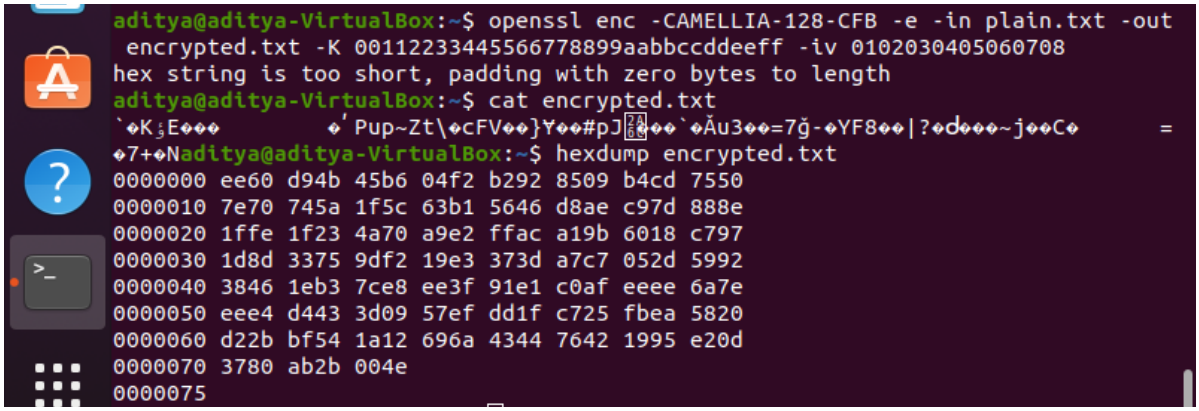


4) Encryption using CAMELLIA-128-ecb

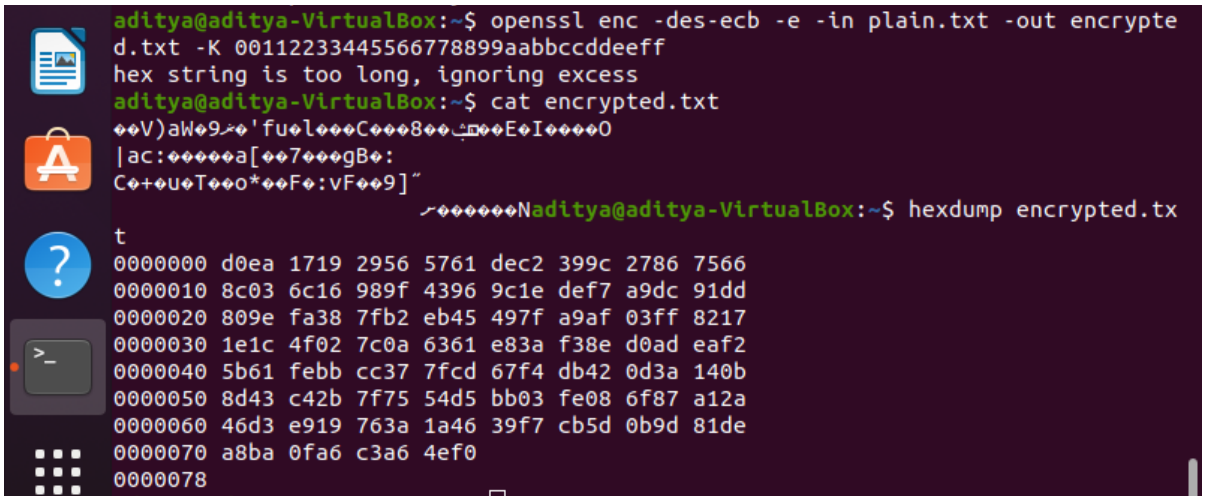


5) Encryption using CAMELLIA-128-cbc

6) Encryption using CAMELLIA-128-cfb

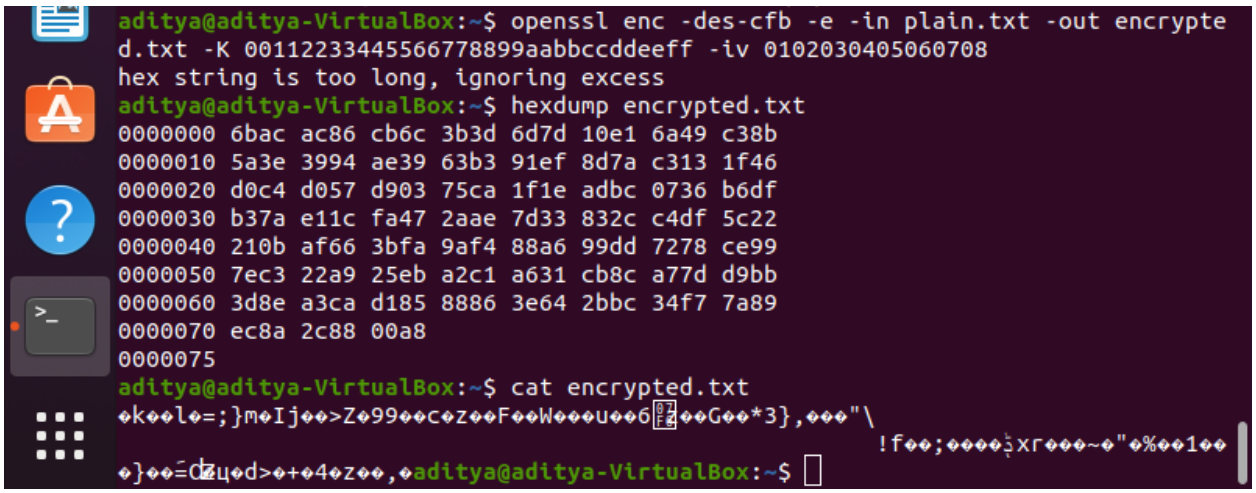


7) Encryption using des-128-ecb

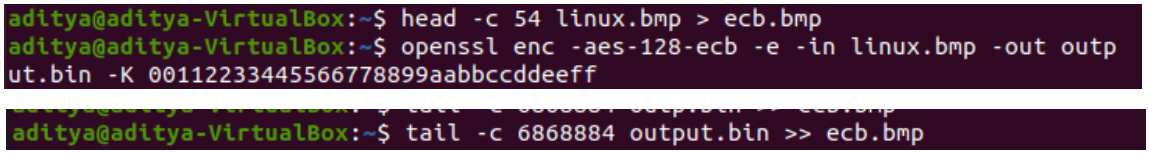


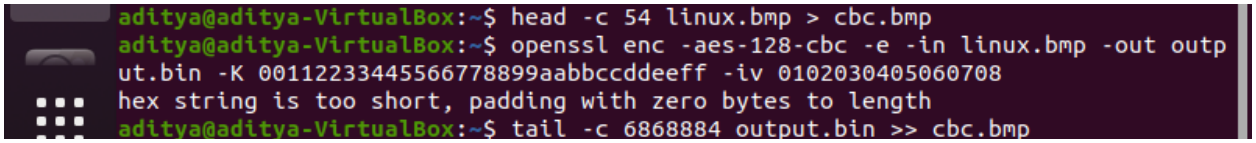
8) Encryption using des-128-cbc

9) Encryption using des-128-cbc

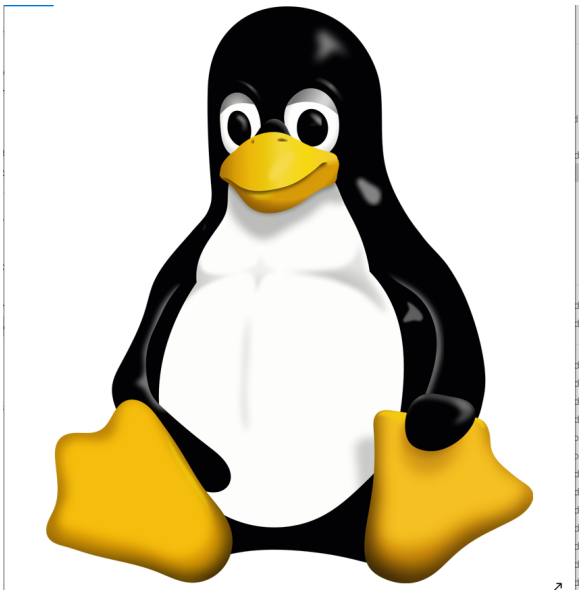


Task 2:

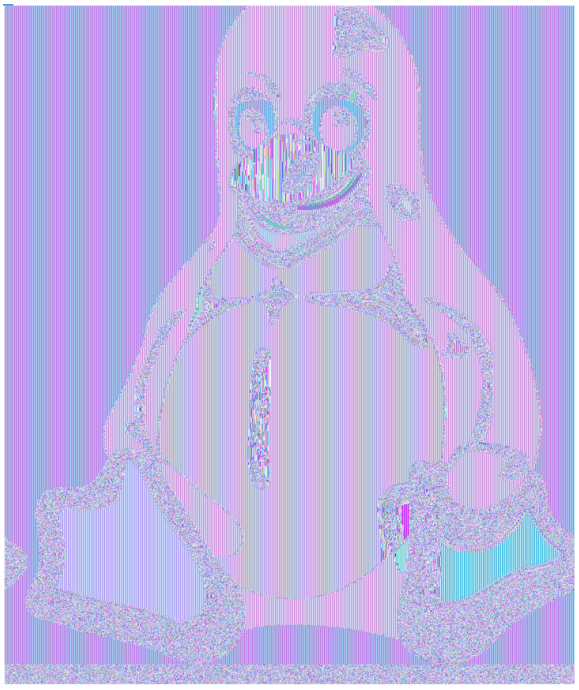




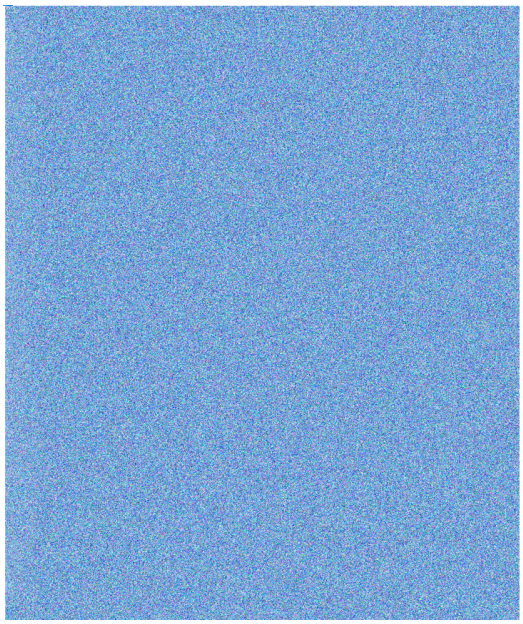
1) Original Image: linux.bmp



2) ECB Encrypted Image: ecb.bmp



3) CBC Encrypted Image: cbc.bmp



Task 3:

1) ECB

```
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ecb -e -in plain.t
xt -out encrypted.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~$ ghex
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ecb -d -in encrypt
ed.txt -out decrypted.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~$ cat decrypted.txt
Houston do you c◆T◆fH◆r◆VU◆t◆◆l◆fidential information and will be e
ncrypted
Attack at homeland in T-30, prepare NOW!
```

2) CBC

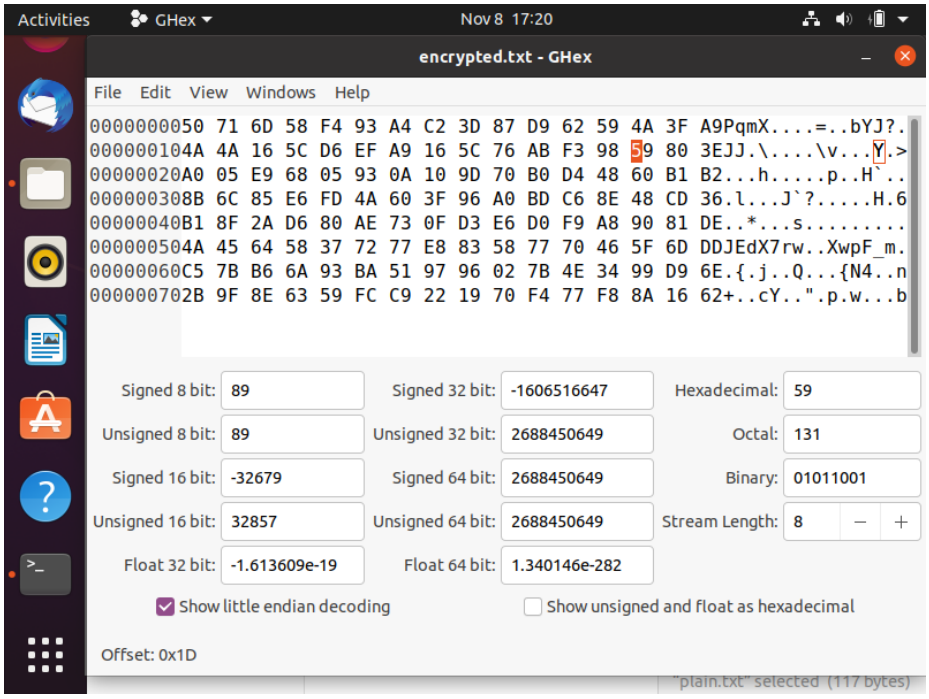
```
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cbc -e -in plain.t
xt -out encrypted.txt -K 00112233445566778899aabbccddeeff -iv 01020
30405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ ghex
aditya@aditya-VirtualBox:~$ cat encrypted.txt
PqmX◆◆◆◆=◆◆bYJ?◆JJ\◆◆\v◆◆◆◆>◆◆h◆
◆p◆◆H`◆◆◆l◆◆◆J`?◆◆◆∃H◆6◆◆*∏◆s◆◆◆◆◆◆◆◆JEdX7rw◆XwpF_m◆◆{◆j◆◆Q◆◆{N4◆◆n
+◆◆cY◆◆"p◆w◆◆baditya@aditya-VirtualBox:~$
aditya@aditya-VirtualBox:~$ ghex
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cbc -d -in encrypt
ed.txt -out decrypted.txt -K 00112233445566778899aabbccddeeff -iv 0
102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat decrypted.txt
Houston do you c:+97Ko◆\◆Q◆fidential inf◆rmation and will be encryp
ted
Attack at homeland in T-30, prepare NOW!
```

3) CFB

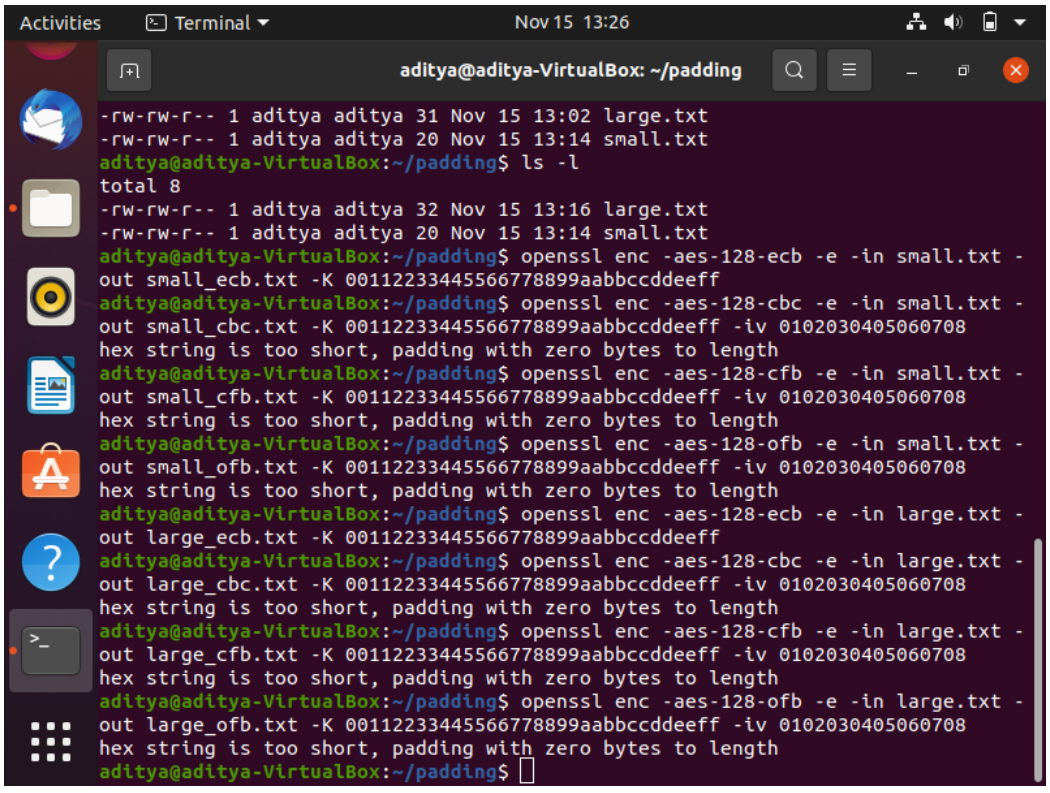
```
ypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ ghex
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cfb -d -in plain.txt -out encr
ypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cfb -e -in plain.txt -out encr
ypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ ghex encrypted.txt
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cfb -d -in encrypted.txt -out
decrypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat decrypted.txt
Houston do you copy?
This iN Con¯◆◆Zq◆=◆O+◆Q◆◆ation and will be encrypted
Attack at homeland in T-30, prepare NOW!
```

4) OFB

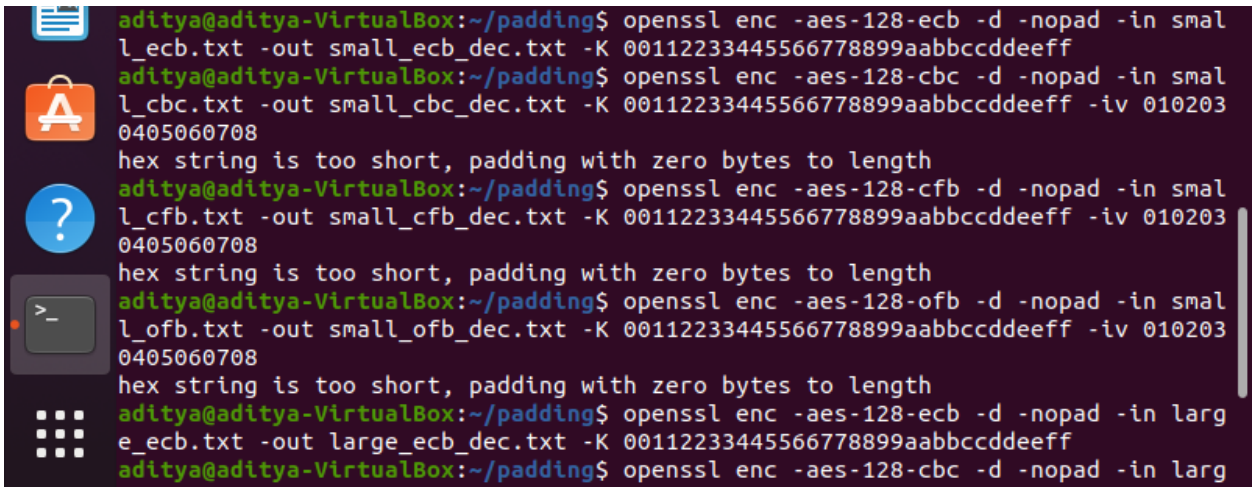
```
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ofb -e -in plain.txt -out encr
ypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ ghex encrypted.txt
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ofb -d -in encrypted.txt -out
decrypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat decrypted.txt
Houston do you copy?
This i6 Confidential information and will be encrypted
Attack at homeland in T-30, prepare NOW!
aditya@aditya-VirtualBox:~$
```

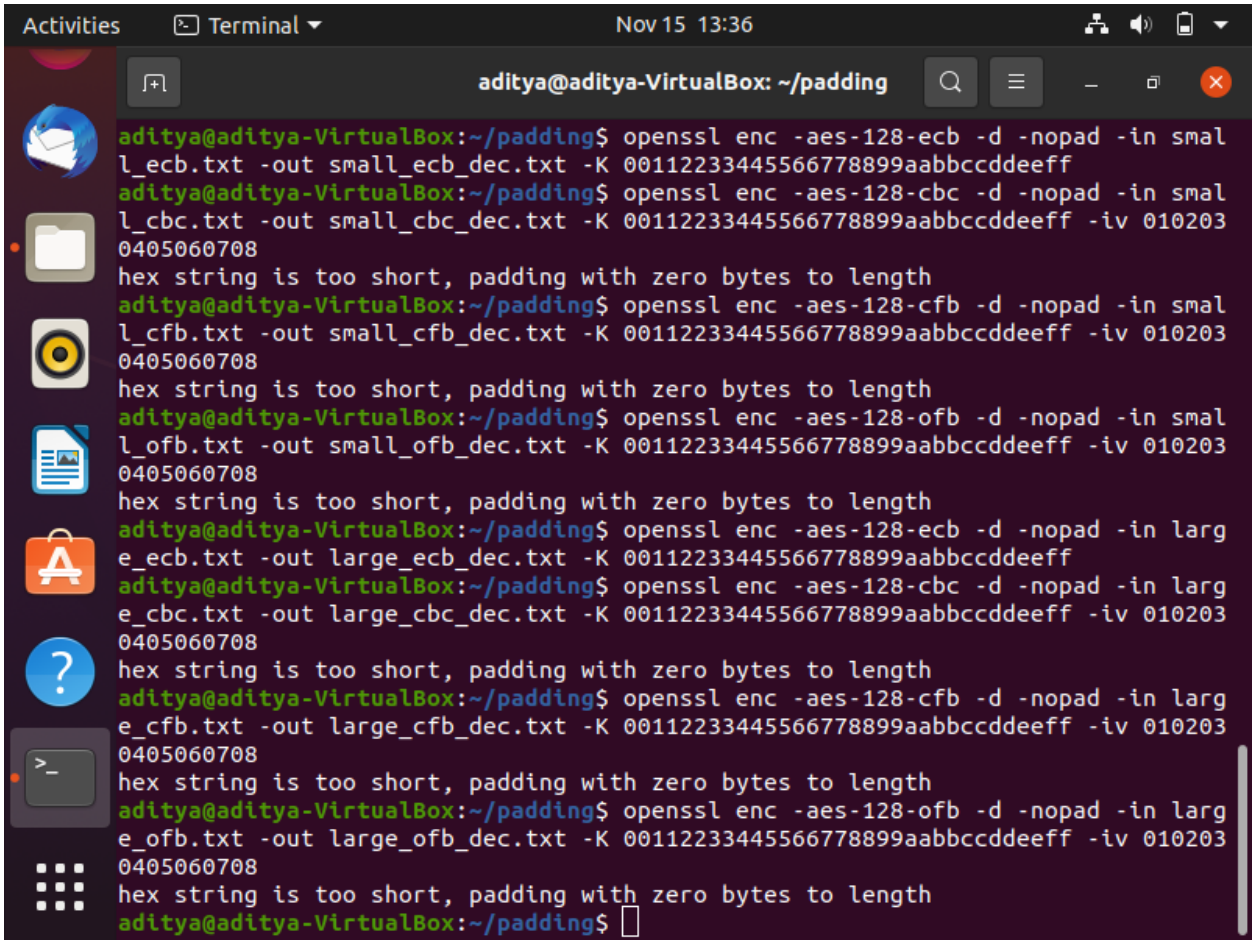

Activities
GHex
Nov 8 17:20
encrypted.txt - GHex
File Edit View Windows Help
00000000 50 71 6D 58 F4 93 A4 C2 3D 87 D9 62 59 4A 3F A9 PqmX....=..bYJ?.
00000010 4A 4A 16 5C D6 EF A9 16 5C 76 AB F3 98 59 80 3E JJ.\....\v...Y.>
00000020 A0 05 E9 68 05 93 0A 10 9D 70 B0 D4 48 60 B1 B2 ...h.....p..H`..
00000030 8B 6C 85 E6 FD 4A 60 3F 96 A0 BD C6 8E 48 CD 36 .l...J`?.....H.6
00000040 B1 8F 2A D6 80 AE 73 0F D3 E6 D0 F9 A8 90 81 DE ..*...s.........
00000050 4A 45 64 58 37 72 77 E8 83 58 77 70 46 5F 6D DD JEdX7rw..XwpF_m.
00000060 C5 7B B6 6A 93 BA 51 97 96 02 7B 4E 34 99 D9 6E .{.j..Q...{N4..n
00000070 2B 9F 8E 63 59 FC C9 22 19 70 F4 77 F8 8A 16 62 +..cY..".p.w...b
Signed 8 bit: 89
Signed 32 bit: -1606516647
Hexadecimal: 59
Unsigned 8 bit: 89
Unsigned 32 bit: 2688450649
Octal: 131
Signed 16 bit: -32679
Signed 64 bit: 2688450649
Binary: 01011001
Unsigned 16 bit: 32857
Unsigned 64 bit: 2688450649
Stream Length: 8
Float 32 bit: -1.613609e-19
Float 64 bit: 1.340146e-282
Show little endian decoding
Show unsigned and float as hexadecimal
Offset: 0x1D
"plain.txt" selected (117 bytes)

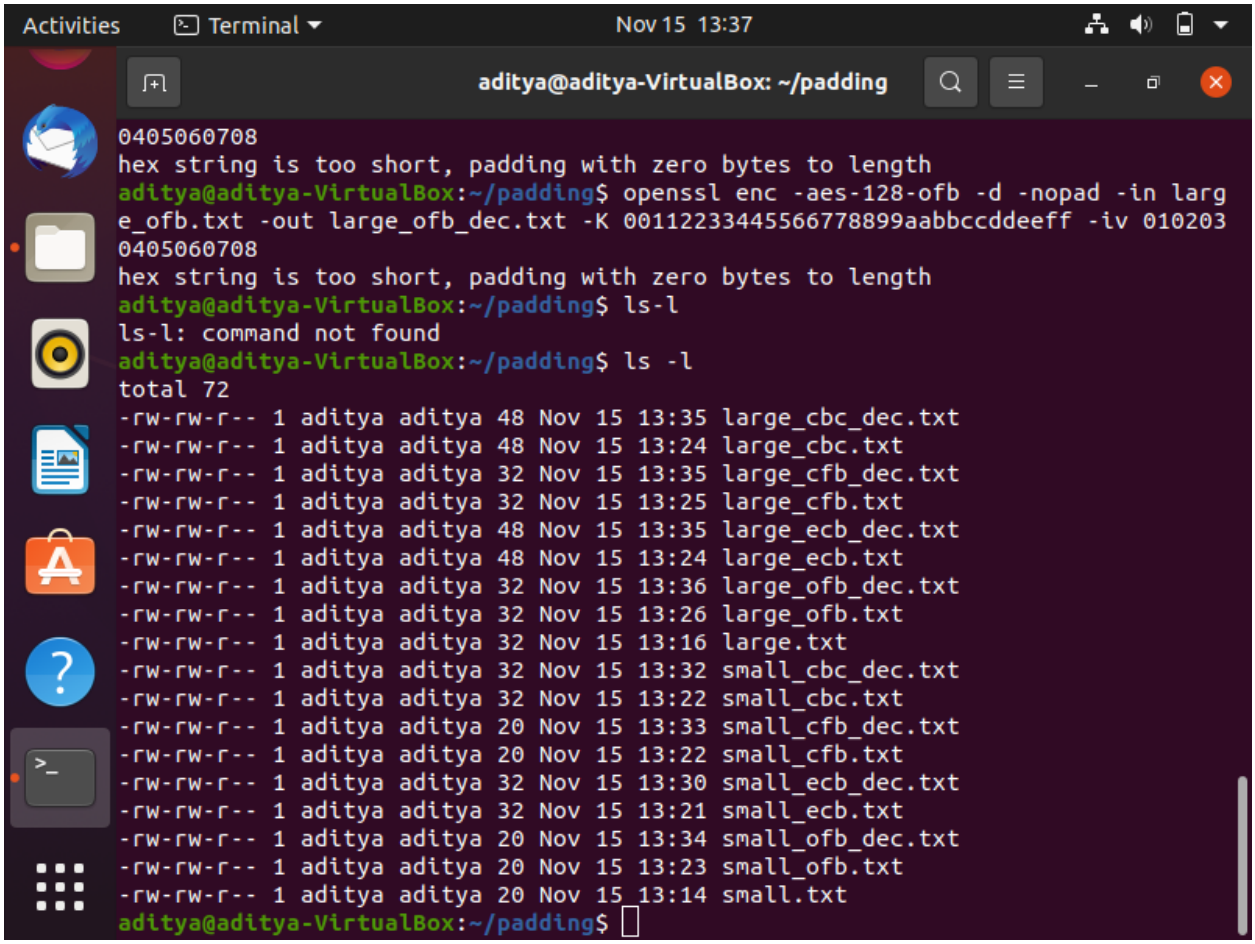
Task 4:


Activities
Terminal
Nov 15 13:26
aditya@aditya-VirtualBox: ~/padding
-rw-rw-r-- 1 aditya aditya 31 Nov 15 13:02 large.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:14 small.txt
aditya@aditya-VirtualBox:~/padding$ ls -l
total 8
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:16 large.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:14 small.txt
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -e -in small.txt -out small_ecb.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -e -in small.txt -out small_cbc.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -e -in small.txt -out small_cfb.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -e -in small.txt -out small_ofb.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -e -in large.txt -out large_ecb.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -e -in large.txt -out large_cbc.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -e -in large.txt -out large_cfb.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -e -in large.txt -out large_ofb.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$


aditya@aditya-VirtualBox:~/padding$ ls -l
total 40
-rw-rw-r-- 1 aditya aditya 48 Nov 15 13:24 large_cbc.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:25 large_cfb.txt
-rw-rw-r-- 1 aditya aditya 48 Nov 15 13:24 large_ecb.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:26 large_ofb.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:16 large.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:22 small_cbc.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:22 small_cfb.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:21 small_ecb.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:23 small_ofb.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:14 small.txt
aditya@aditya-VirtualBox:~/padding$


aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -d -nopad -in smal
l_ecb.txt -out small_ecb_dec.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -d -nopad -in smal
l_cbc.txt -out small_cbc_dec.txt -K 00112233445566778899aabbccddeeff -iv 010203
0405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -d -nopad -in smal
l_cfb.txt -out small_cfb_dec.txt -K 00112233445566778899aabbccddeeff -iv 010203
0405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -d -nopad -in smal
l_ofb.txt -out small_ofb_dec.txt -K 00112233445566778899aabbccddeeff -iv 010203
0405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -d -nopad -in larg
e_ecb.txt -out large_ecb_dec.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -d -nopad -in larg


Activities
Terminal
Nov 15 13:36
aditya@aditya-VirtualBox: ~/padding
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -d -nopad -in small_ecb.txt -out small_ecb_dec.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -d -nopad -in small_cbc.txt -out small_cbc_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -d -nopad -in small_cfb.txt -out small_cfb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -d -nopad -in small_ofb.txt -out small_ofb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -d -nopad -in large_ecb.txt -out large_ecb_dec.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -d -nopad -in large_cbc.txt -out large_cbc_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -d -nopad -in large_cfb.txt -out large_cfb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -d -nopad -in large_ofb.txt -out large_ofb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$



Task 5:

1) Code:

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
plain_text_inbytes = b"This is a top secret."
cipher_hex =
"8d20e5056a8d24d0462ce74e4904c1b513e10d1df4a2ef2ad4540fae1ca0aaf9"
answer=""
file = open('words.txt', 'r')
raw_text = file.readlines()
words = [str.strip(line) for line in raw_text]
# print(plain_text_inbytes,cipher_hex)
# print(raw_text)
# print(words)
```

```
for word in words:
    if len(word) >= 16:
        continue
    word = word.lower()
    key = word.encode() + b' '*(16-len(word))
    cipher = AES.new(key, AES.MODE_CBC, iv=bytes.fromhex('0'*32))
    ciphertext = cipher.encrypt(pad(plain_text_inbytes, AES.block_size))
    if bytes.hex(ciphertext) == cipher_hex:
        answer=word
        break
print("Key used to encrypt given plain text:",answer)
```

2) Output:

```
PS C:\Users\adity\Desktop\ADM\College\Coding\Cryptography SS>  c:; cd 'c:\Users\adity\Desktop\ADM\College\Co
ding\Cryptography SS'; & 'C:\Users\adity\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\adity\
.vscode\extensions\ms-python.python-2021.11.1422169775\pythonFiles\lib\python\debugpy\launcher' '54708' '--'
 'c:\Users\adity\Desktop\ADM\College\Coding\Cryptography SS\task5.py'
Key used to encrypt given plain text: median
```

Conclusion:

1) Task 1:

In this task I learned how to use Openssl to encrypt and decrypt messages using various algorithms and modes. I also learned that AES, DES, and CAMELLIA are symmetric key block algorithms that may use the same keys to encrypt and decrypt the data.

2) Task 2:

a) The encrypted image produced by ECB is very similar to the original and is very intuitive to guess what is the original image by just looking at it.
b) CBC on the other hand encrypts the image very well, and one cannot make out what the original image is. Hence we understand that CBC is much better for image encryption than ECB.

3) Task 3:

a) In ECB mode, only one block is affected when a ciphertext problem occurs. As it is decrypted individually. Hence, the corrupted bit in the 30th byte affects the entire block that it is present in.
b) CBC on the other hand depends on the results of XOR of the previous blocks, hence two blocks are affected.
c) In CFB mode, there is a problem with the number of n / r blocks.
r = The number of bits taken at one time to XOR.

d) In OFB mode, one digit of the 30th byte is corrupted, and in plain text only that byte or that character is broken. Therefore, only OFB mode shows the most promising results and almost all the text will be restored.

4) Task 4:

a) I observed the size of CBC and ECB encrypted files with the nopad option is 12 bytes more for the 20 bytes file and 16 bytes larger for the 32 bytes file, however the size of OFB and CFB decrypted files is the same.
b) Hence, we can conclude that ECB and CBC are block ciphers while CFB and OFB are stream ciphers.

5) Task 5:

In this task we understand that with plain text, cipher text, iv and a reasonably large list of possible keys we can find the key by brute force method. Although if the list of possible keys is very large then it is not computationally feasible to do the same. We also learned how to use the pycryptodome library in python to encrypt and decrypt messages.