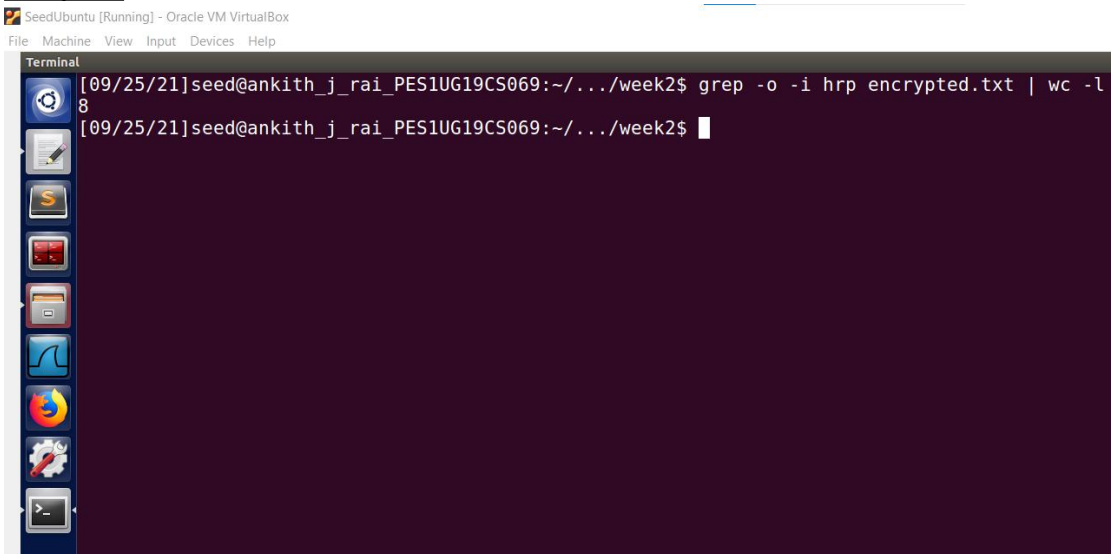
CRYPTOGRAPHY LAB- WEEK 2
Secret Key Encryption

NAME : Ankith J Rai
SRN : PES1UG19CS069
SEC : B

Task 1: Frequency Analysis Against Mono-alphabetic Substitution Cipher

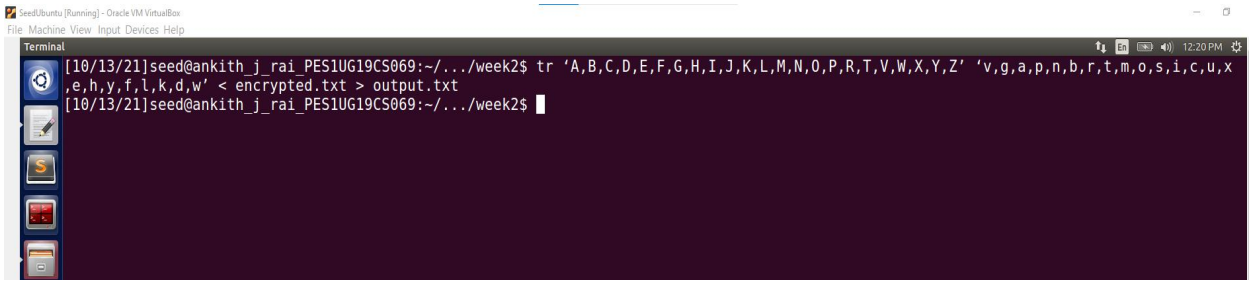
Step 1:



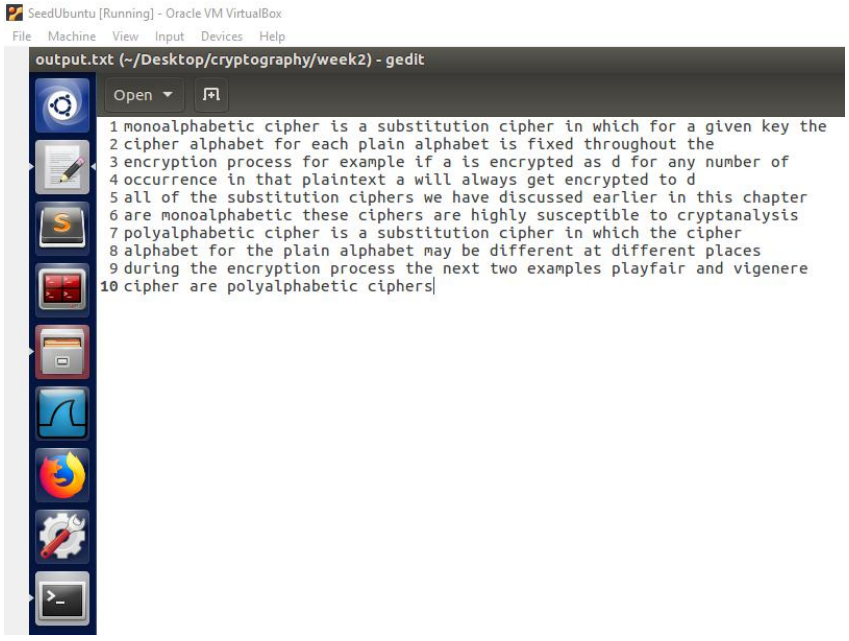
We can see that the number of instances of the word 'HRP' is 8 in the encrypted.txt file.

Step 2: As HRP is frequently occurring trigram hence HRP is replaced with the word 'the'.

After Frequency analysis we get the mapping of the cipher character with that of the plain text character.

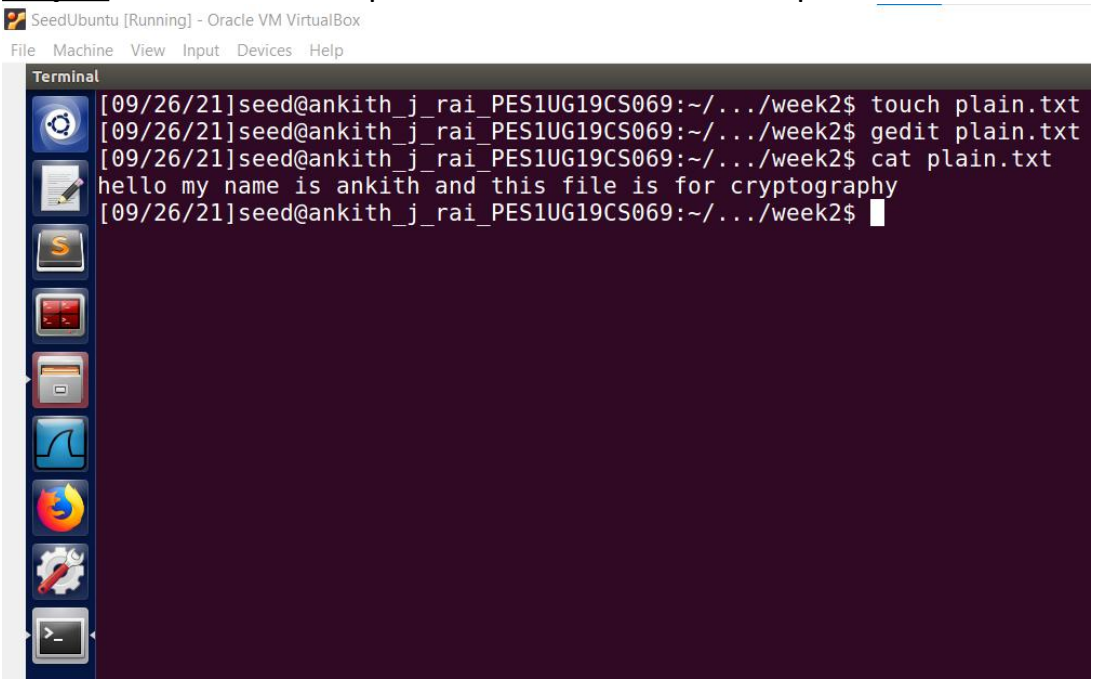


After replacing the cipher character that of the plain text character we can see the plain text which was encrypted below.



Task 2: Encryption using different ciphers and modes.

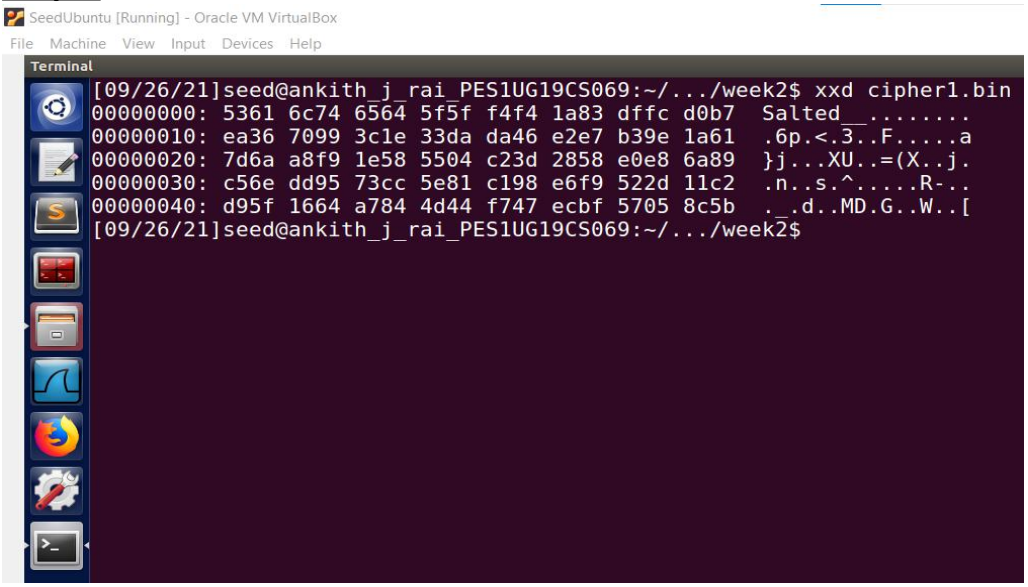
Step 1: First we create a plain.txt file with our own input.



For cipher type = aes-128-cbc

Step2:

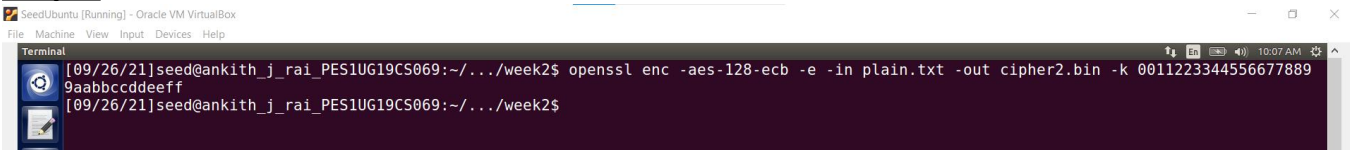
Step3:



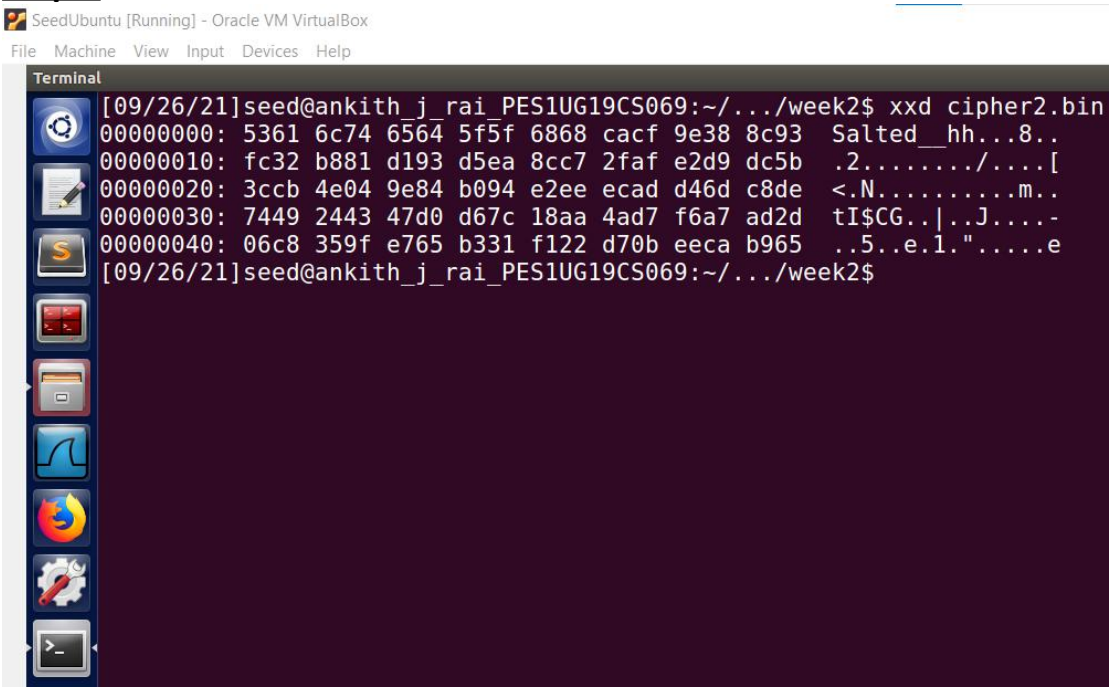
The above screenshot is the cipher text of plain.txt file content for cipher type = aes128-cbc.

For cipher type = aes-128-ecb

Step2:



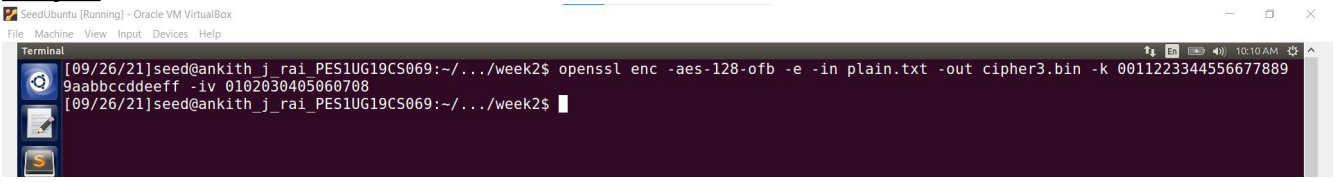
Step3:



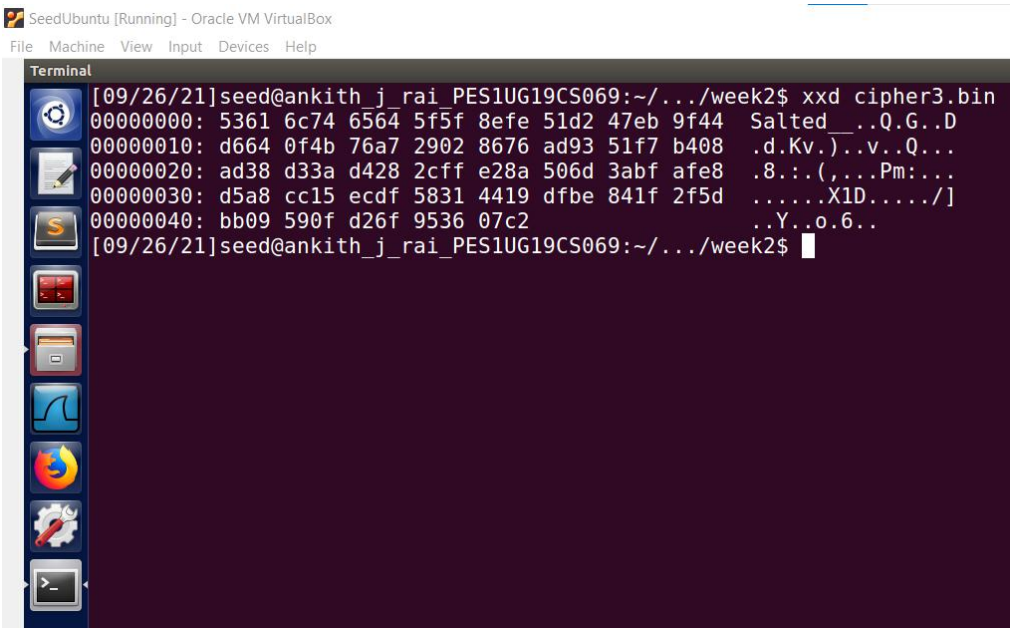
The above screenshot is the cipher text of plain.txt file content for cipher type = aes128-ecb

For cipher type = aes-128-ofb

Step2:



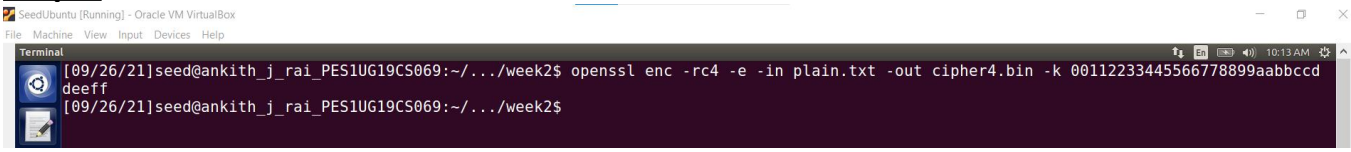
Step3:



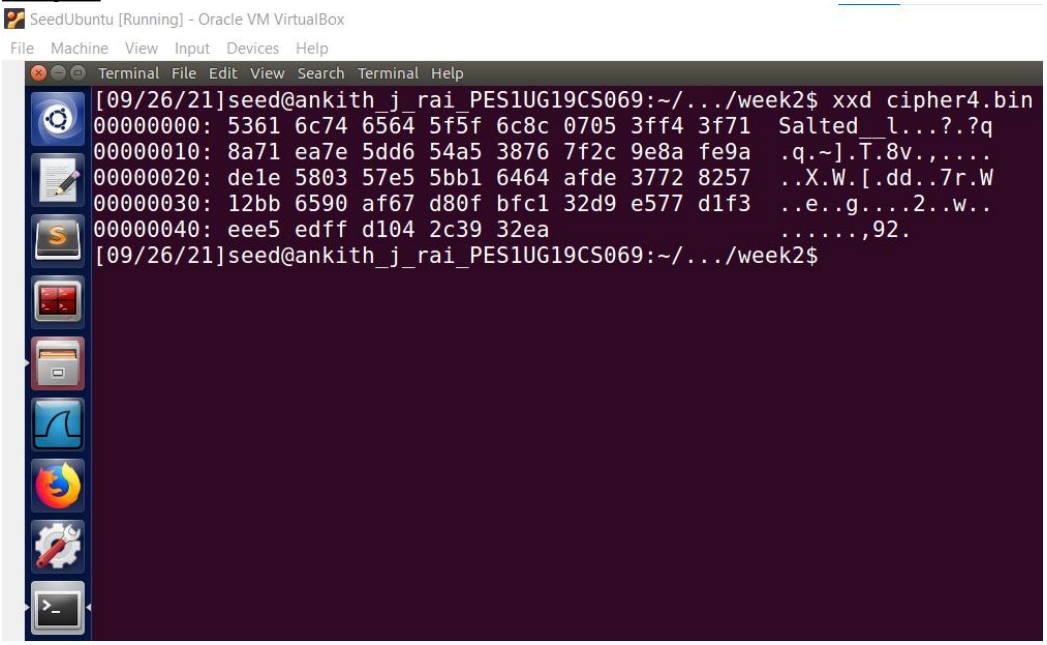
The above screenshot is the cipher text of plain.txt file content for cipher type = aes128-ofb

For cipher type = rc4

Step2:



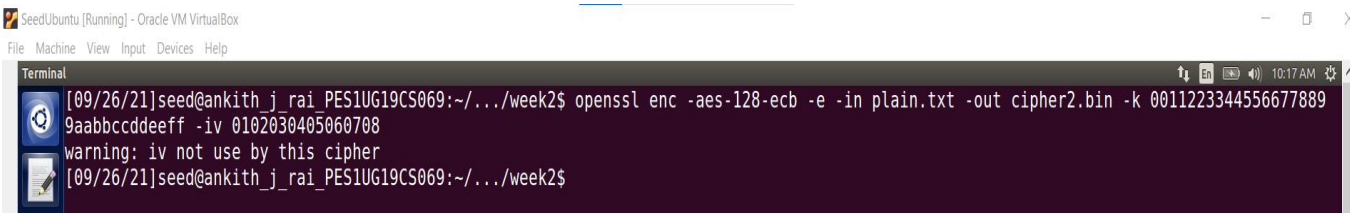
Step3:

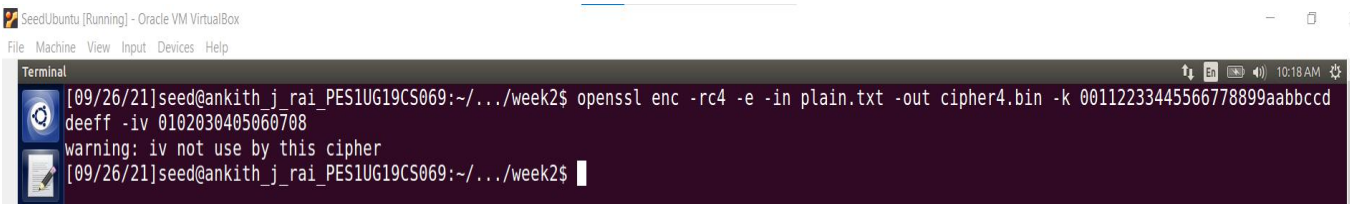


The above screenshot is the cipher text of plain.txt file content for cipher type = rc4

Q1. Which out of the above ciphers and modes do not require an initialization vector?

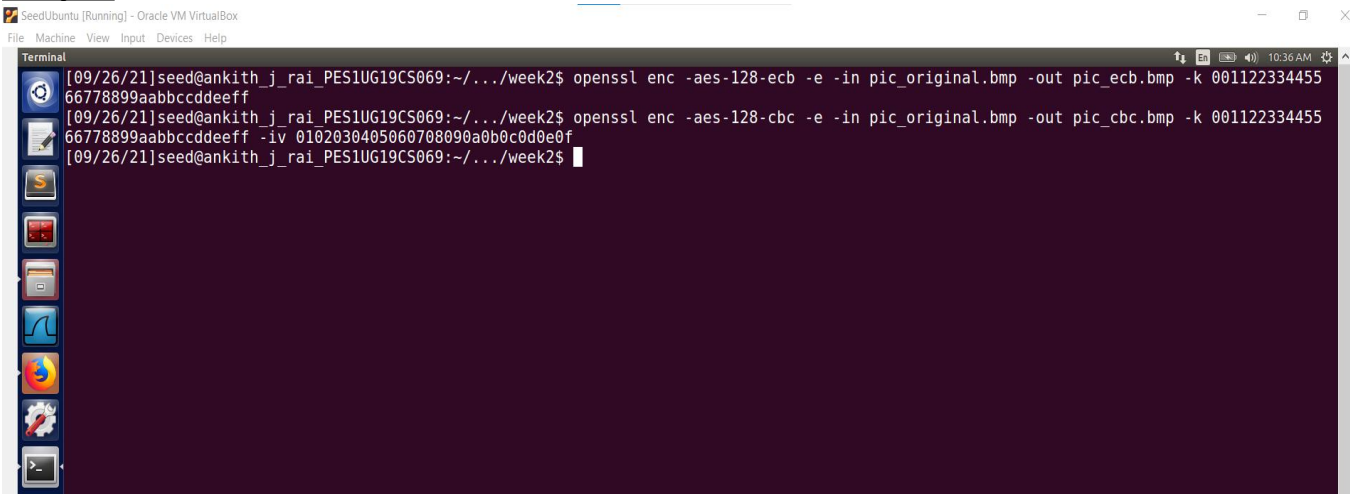
Ans) From doing the above task it was noted that cipher type aes-128-ecb and rc4 do not require a initialization vector.





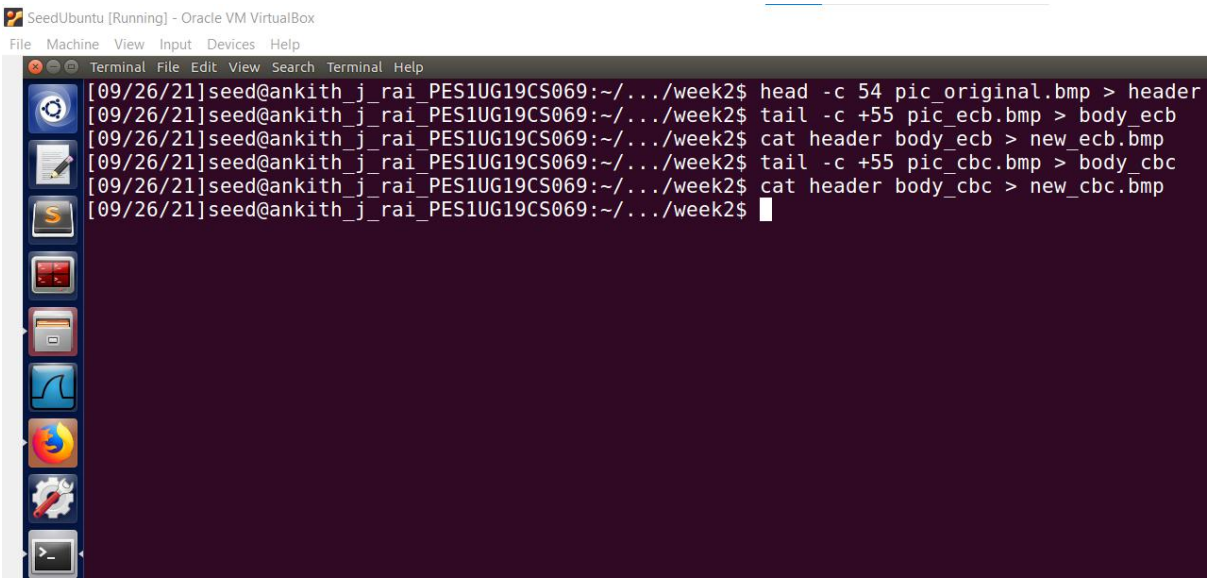
Task 3: Encryption Mode ECB vs CBC

Step1:



Hence from above commands pic_ecb.bmp and pic_cbc.bmp is generated.

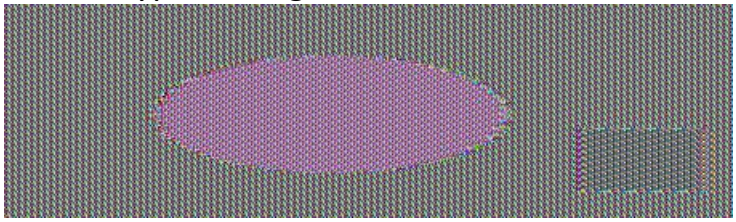
Step2:



Step3:

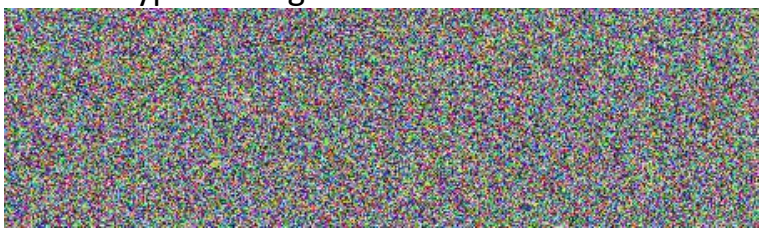
For ecb encryption:

The encrypted image is:



For cbc encryption:

The encrypted image is:

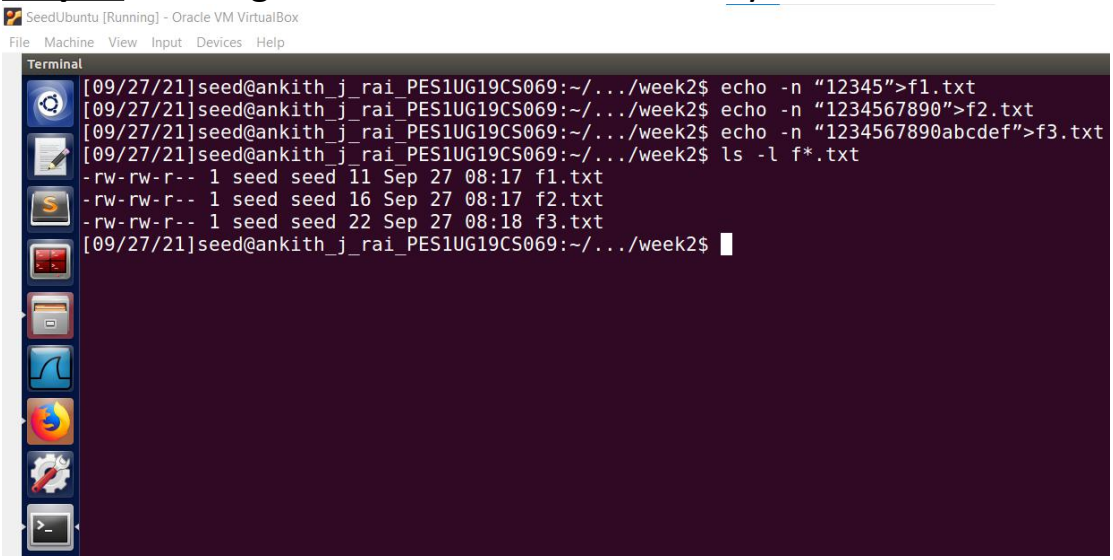


Q)Which out of the two is better and why?

Ans) From the above two encrypted images we can see that the cbc encrypted image is much more protected(i.e better) than a ecb encrypted image.

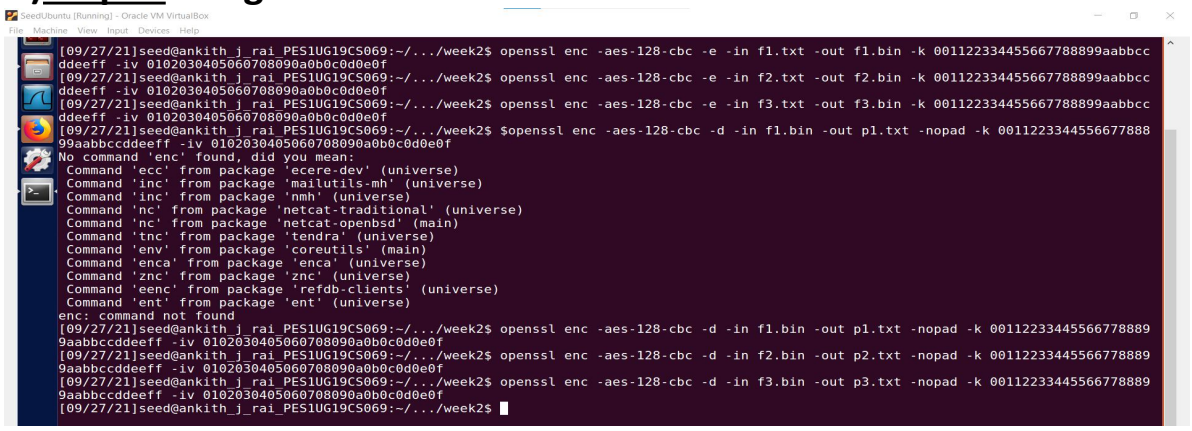
Task 4: Padding

Step1: Creating three files of size 5,10,15 bytes.

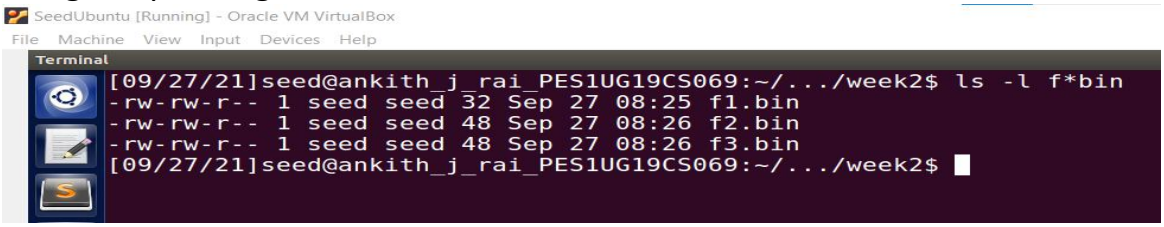


Let file file f1,f2,f3 be files of size 5 bytes,10 bytes and 15 bytes respectively.

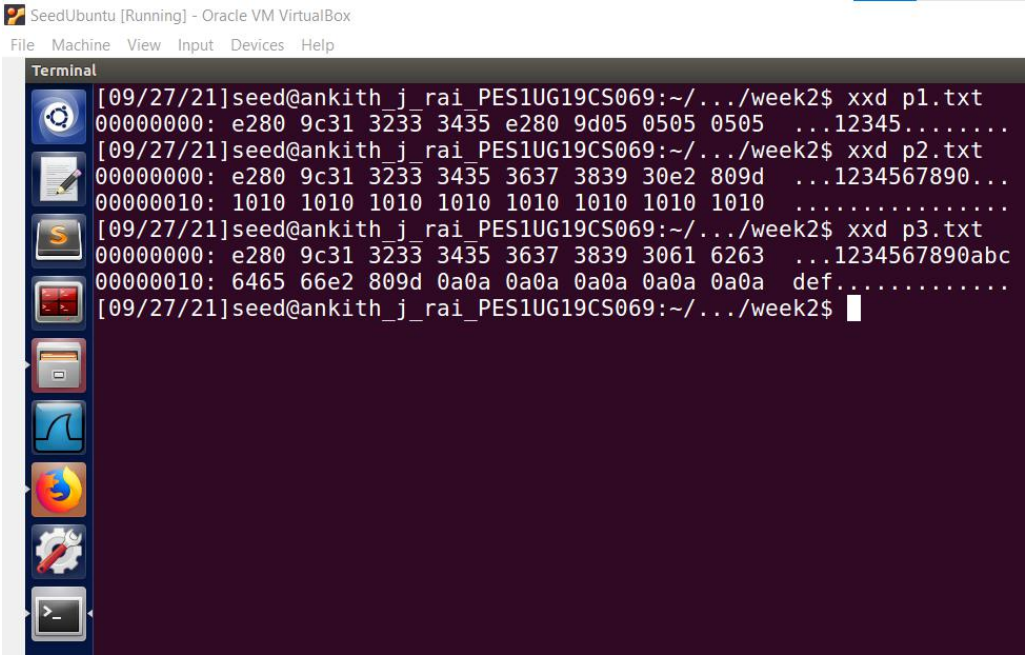
A)Step2: Using cbc mode



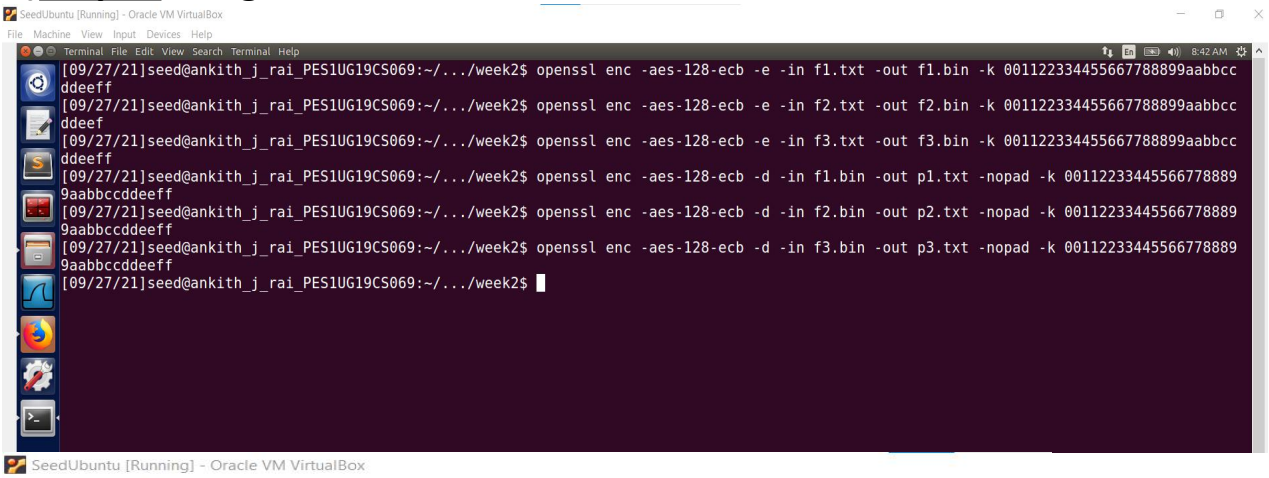
In this step we have encrypted using cbc mode and decrypted without using the padding.

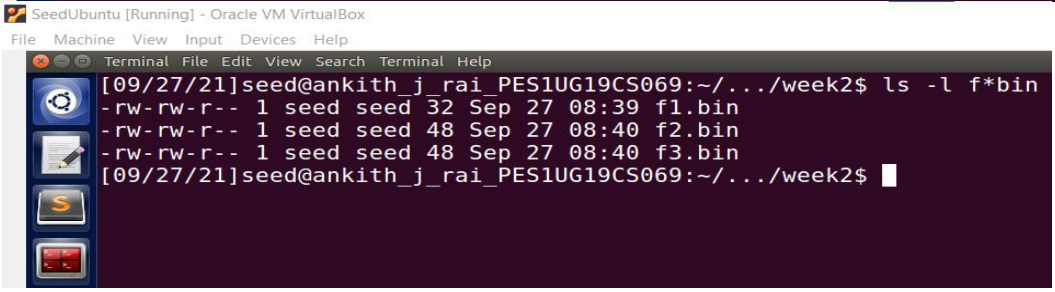


Step3:

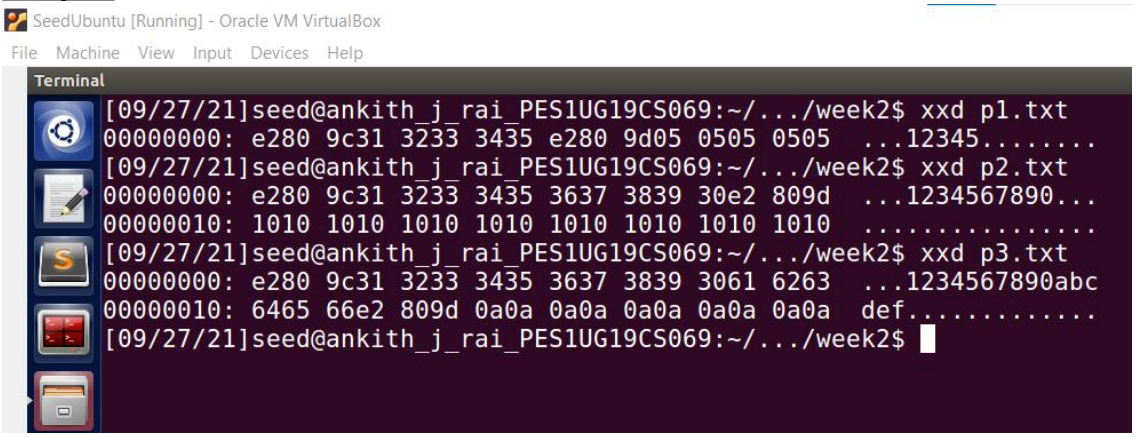


B)Step2: Using ecb mode

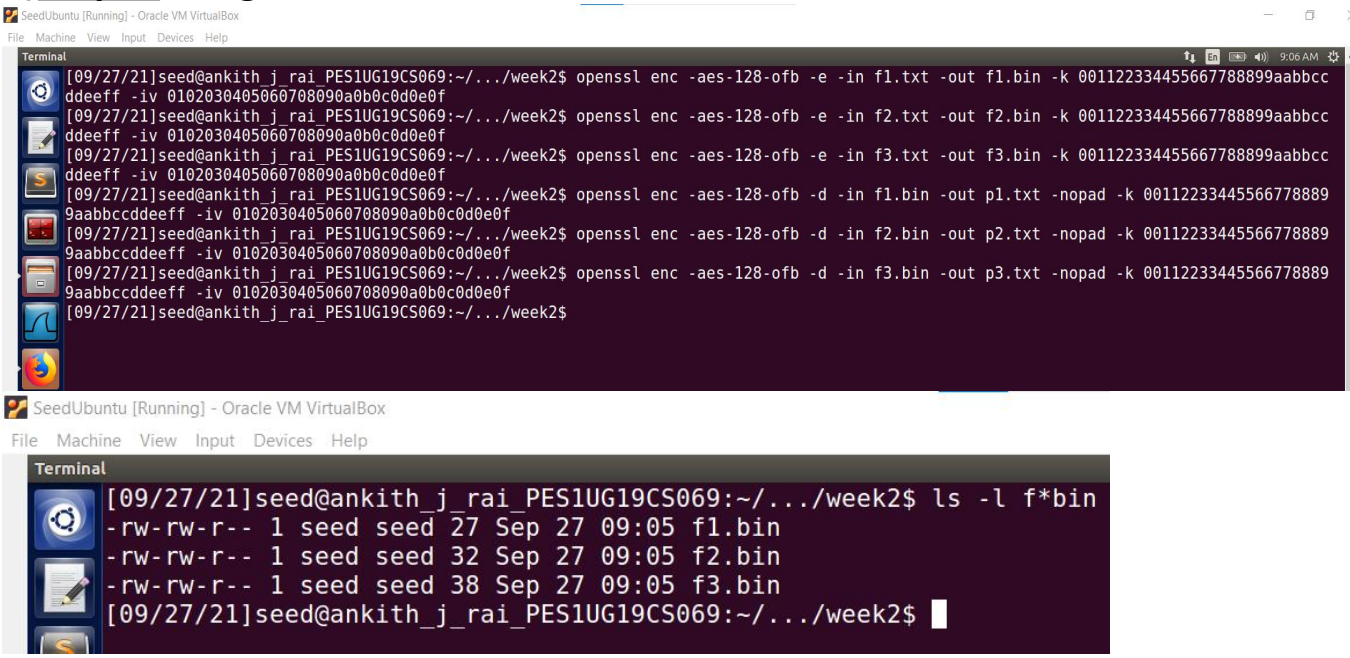




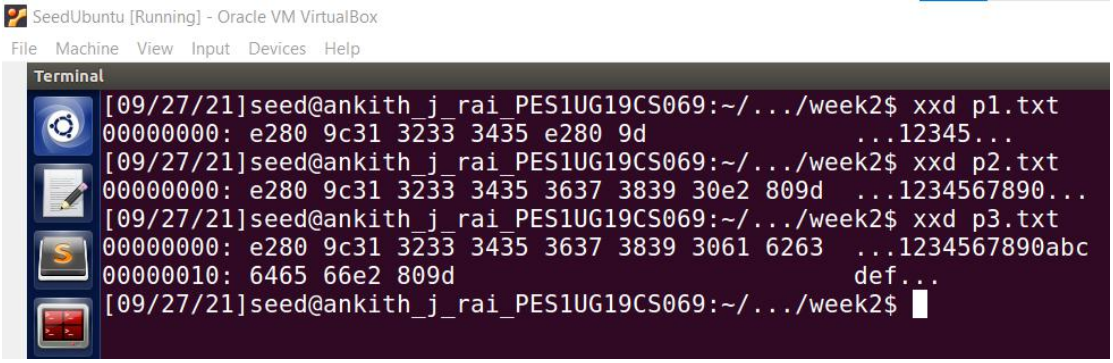
Step3:



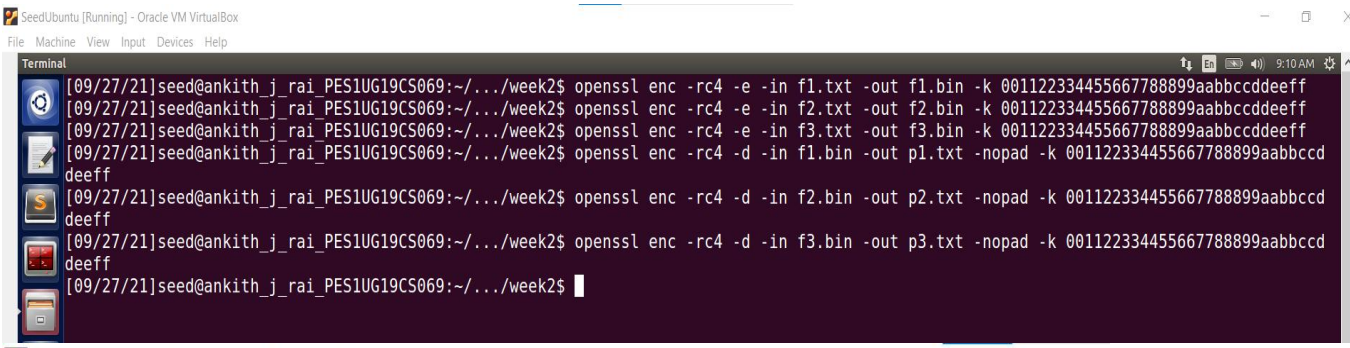
C)Step2: Using ofb mode

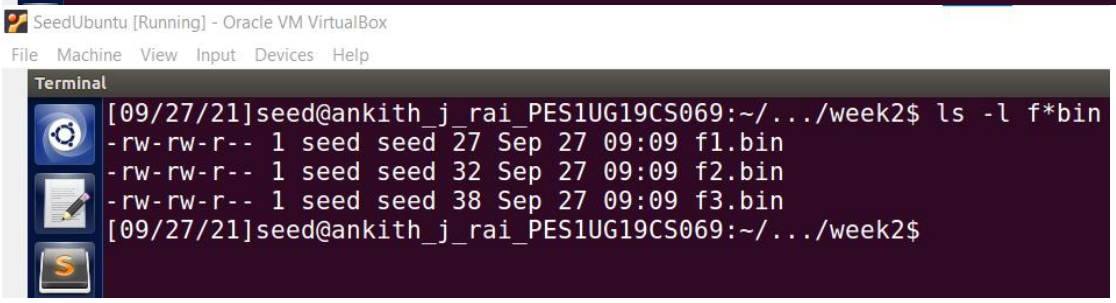


Step3:

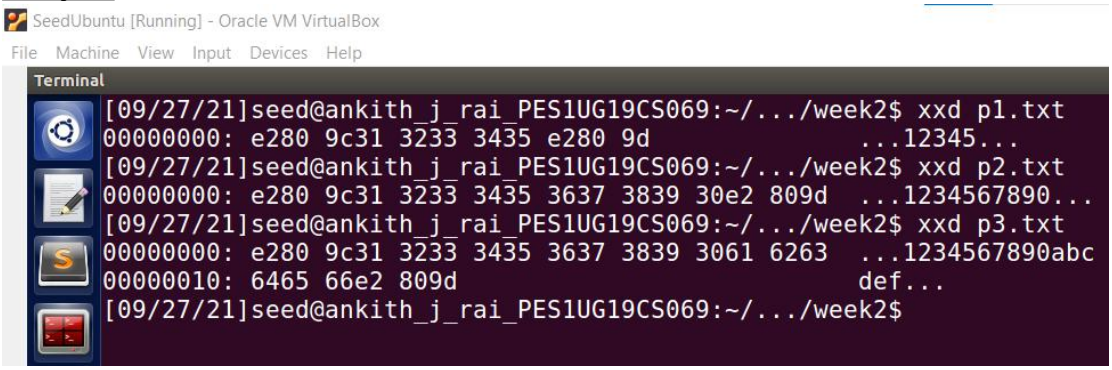


D)Step2: Using rc4 mode



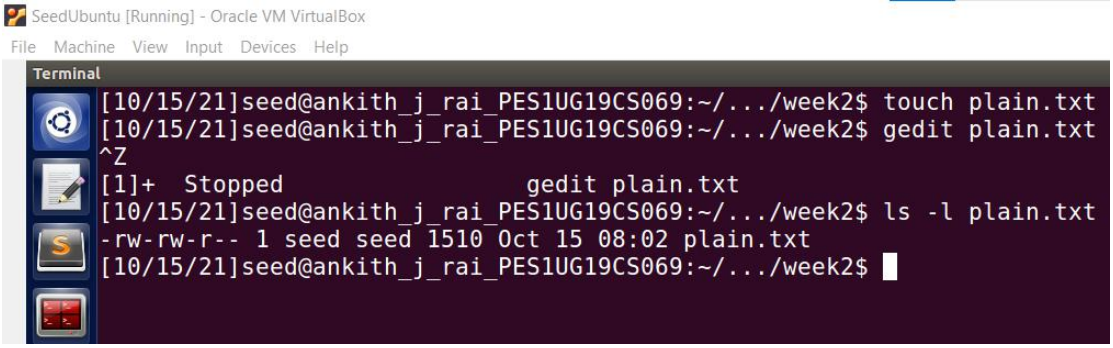


Step3:



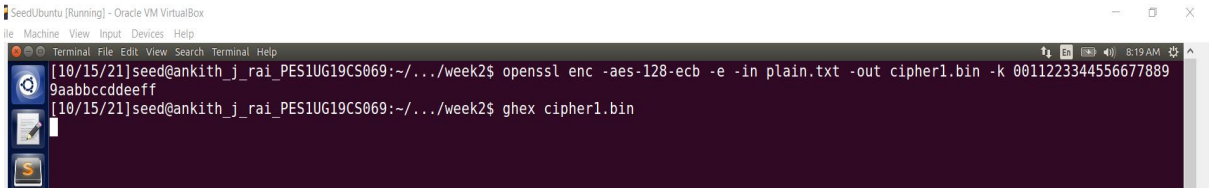
Task 5: Error propagation – corrupted Ciphertext

Step1: Creating a file plain.txt with size more than 1000 bytes.

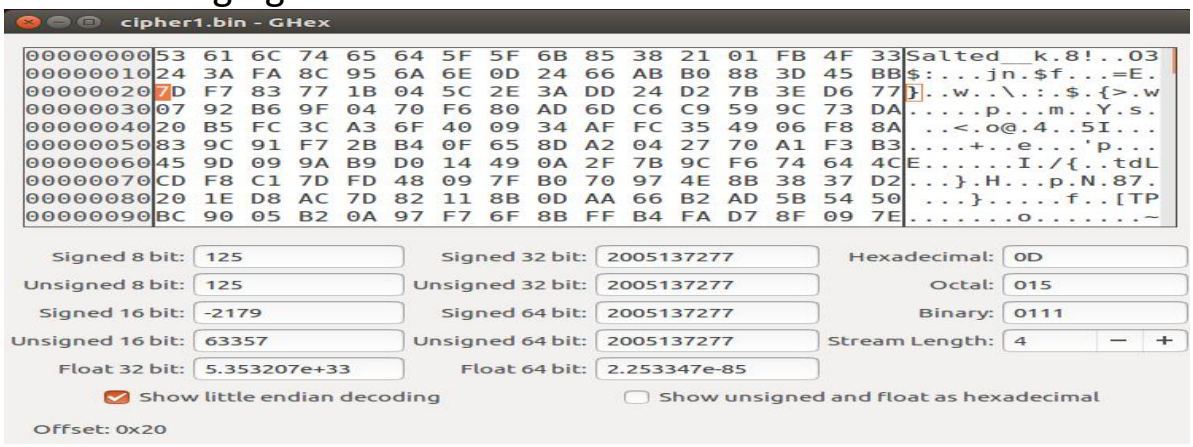


A) For ecb mode

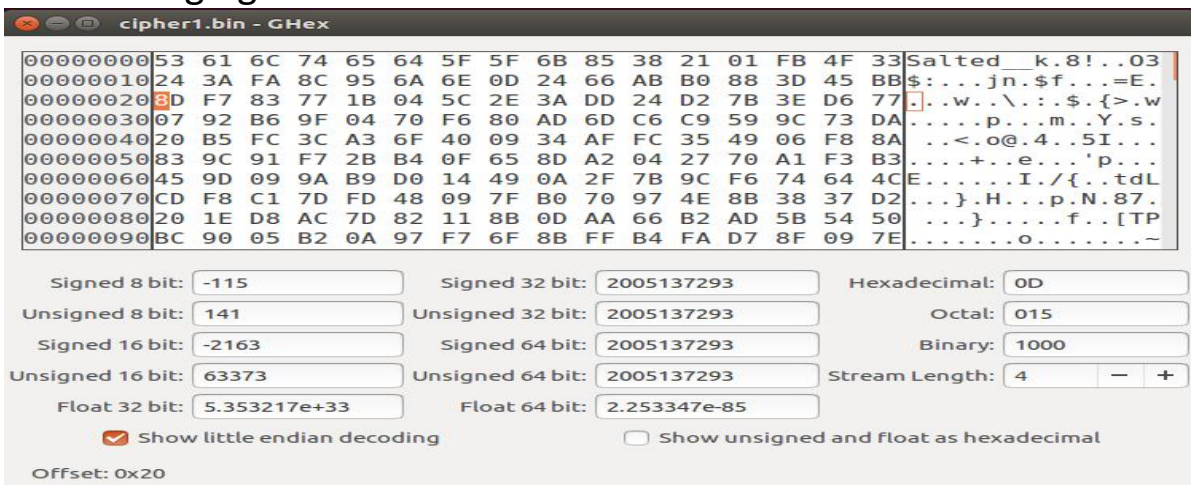
Step2:



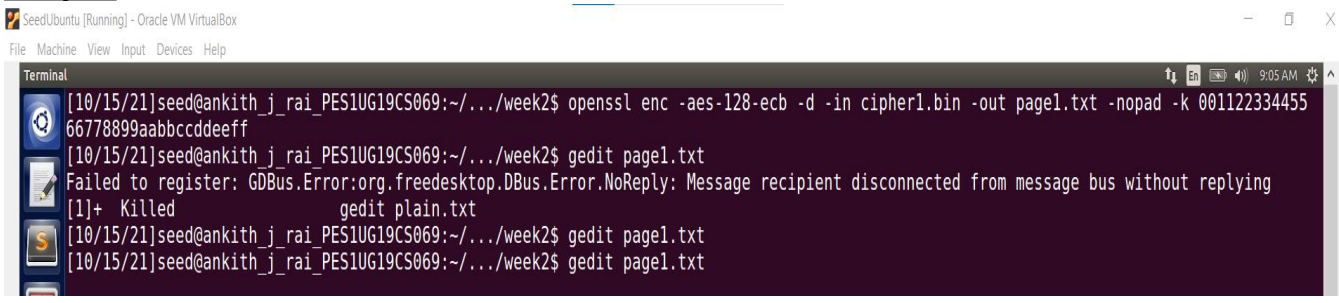
Before changing the bit:

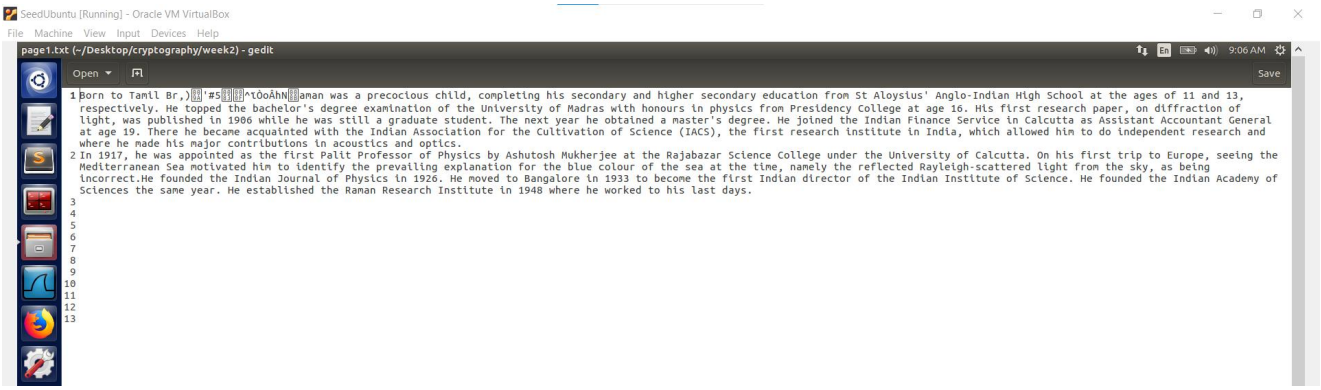


After changing the bit:



Step3:

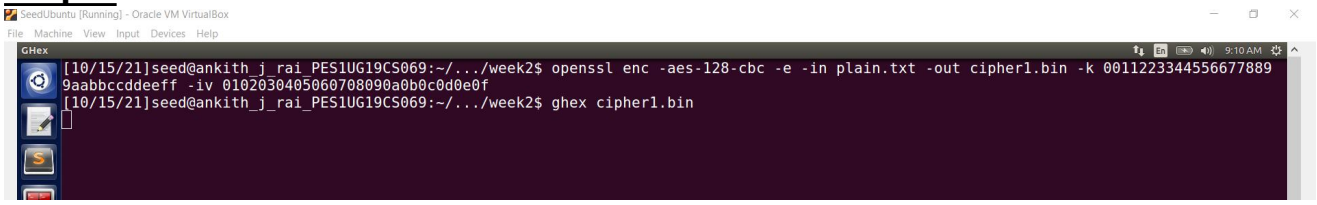




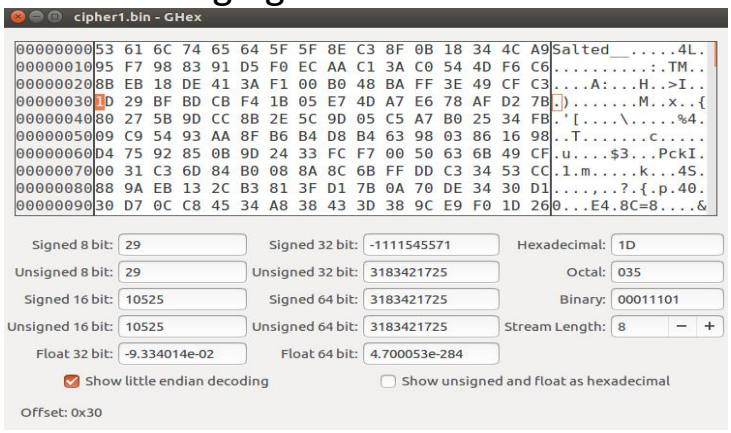
On decrypting Cipher1.bin we can see that the decrypted file has been corrupted.

B) For cbc mode

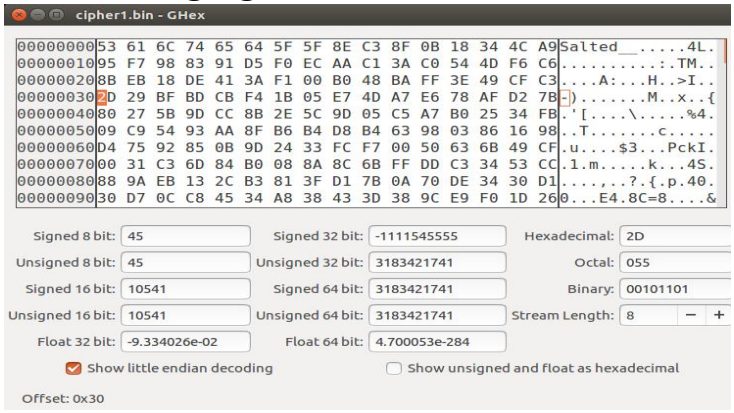
Step2:



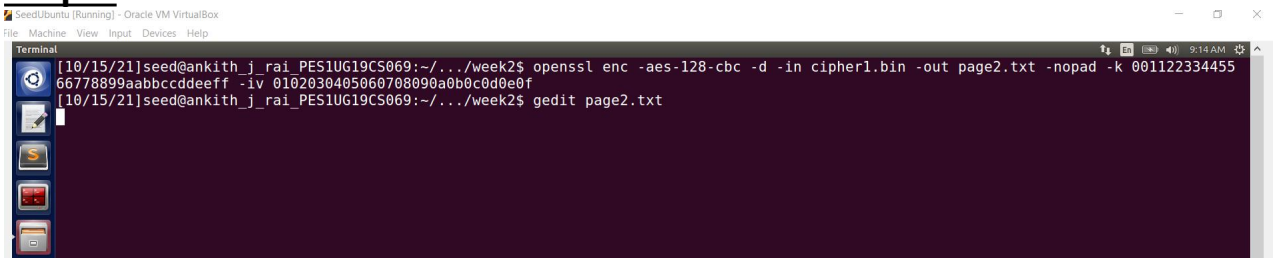
Before changing the bit:

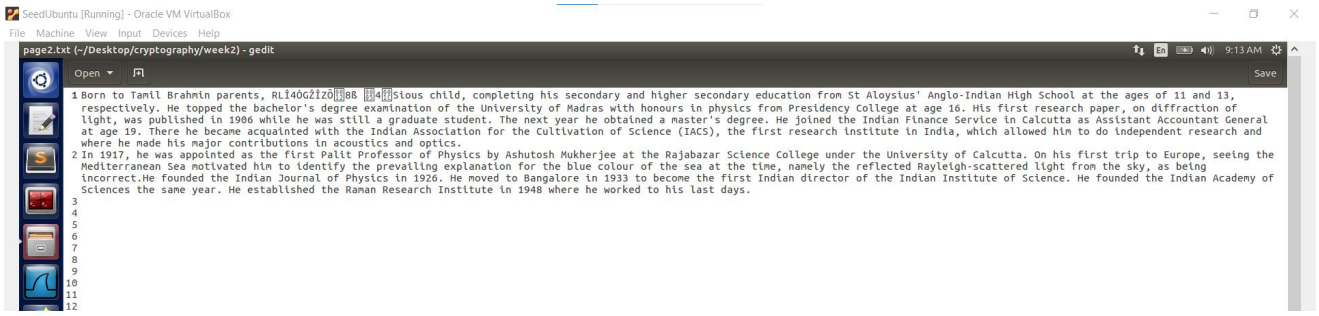


After changing the bit:



Step3:

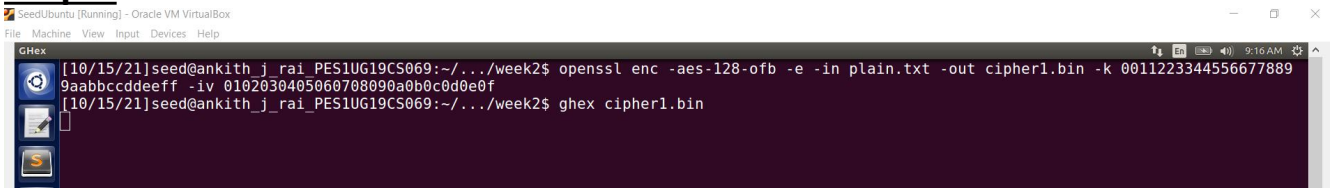




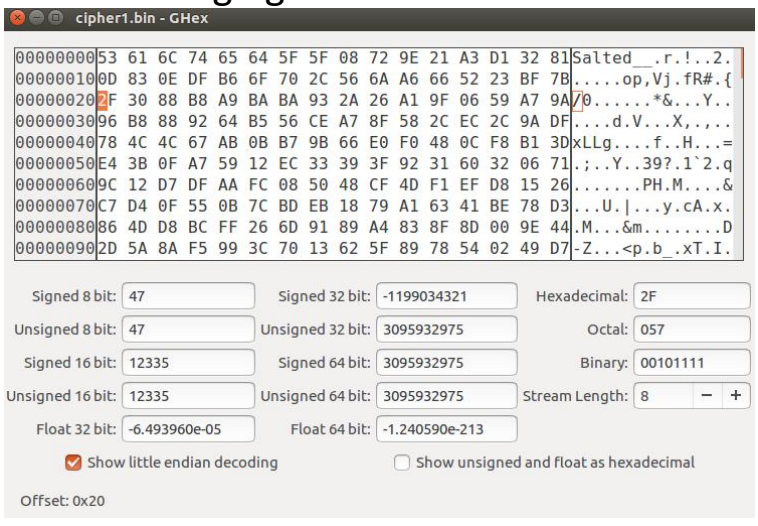
On decrypting Cipher1.bin we can see that the decrypted file has been corrupted.

C) For ofb mode

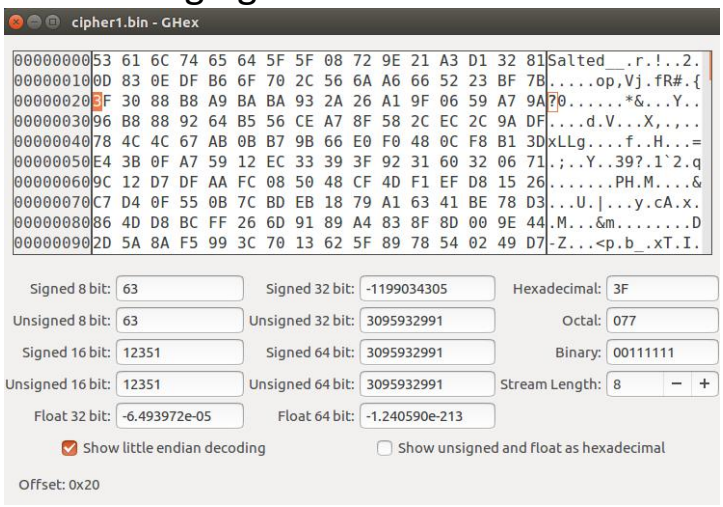
Step2:



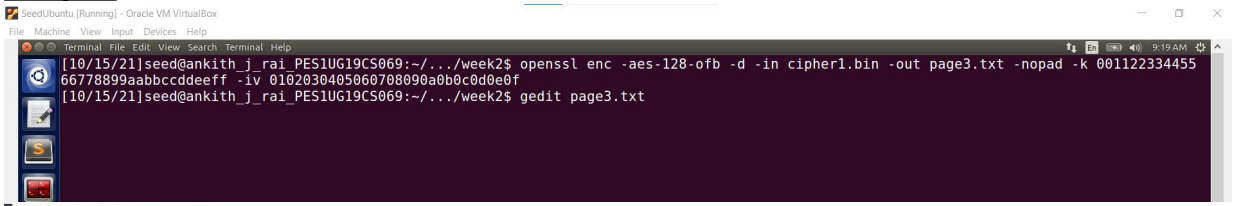
Before changing the bit:

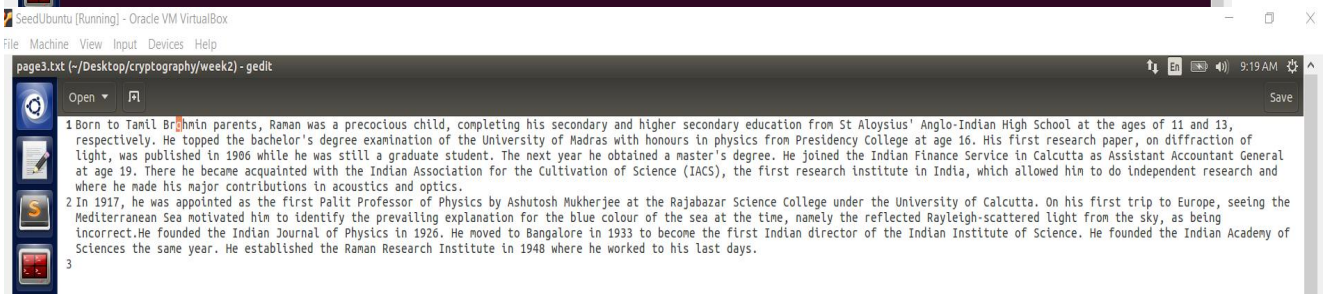


After changing the bit:



Step3:

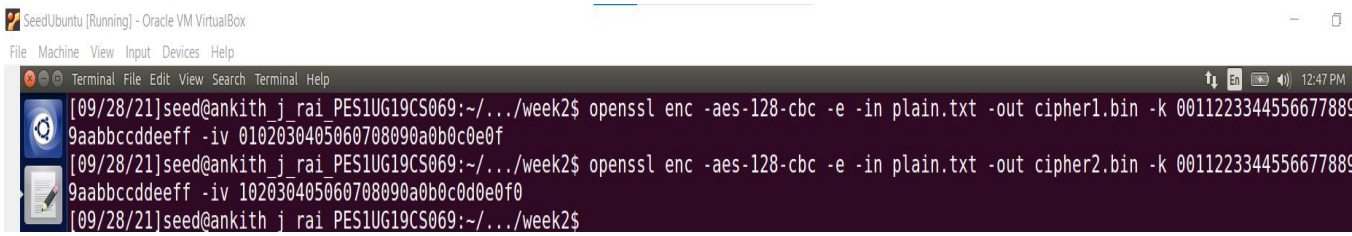




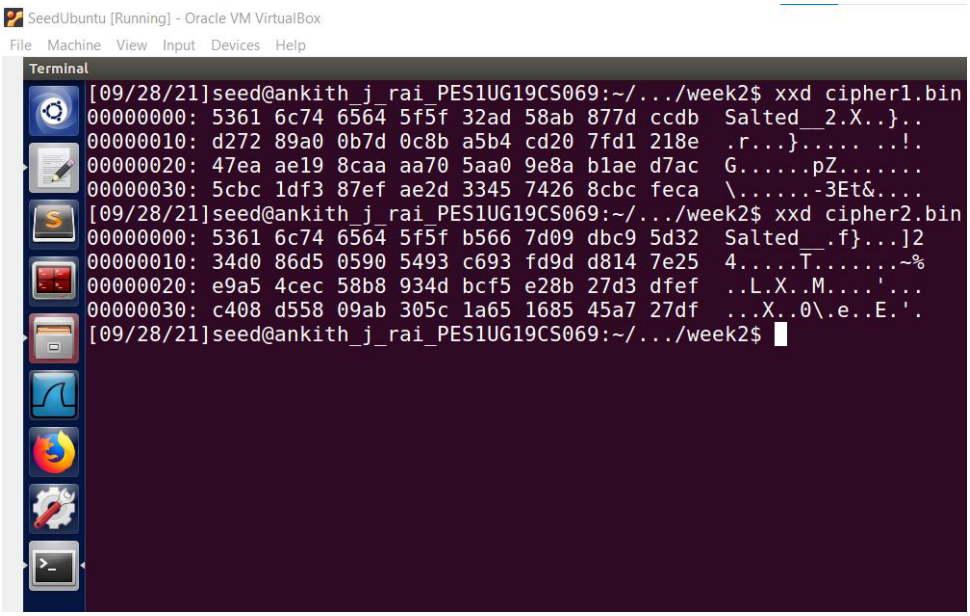
On decrypting Cipher1.bin we can see that the decrypted file has been corrupted as we can see that in the word ‘Brahmin’ in plain text has been changed to ‘Brqhmin’ in the page3.txt in decrypted file.

Task 6: Initialization Vector

Step 1: Encrypting the same plain text using different initial vector

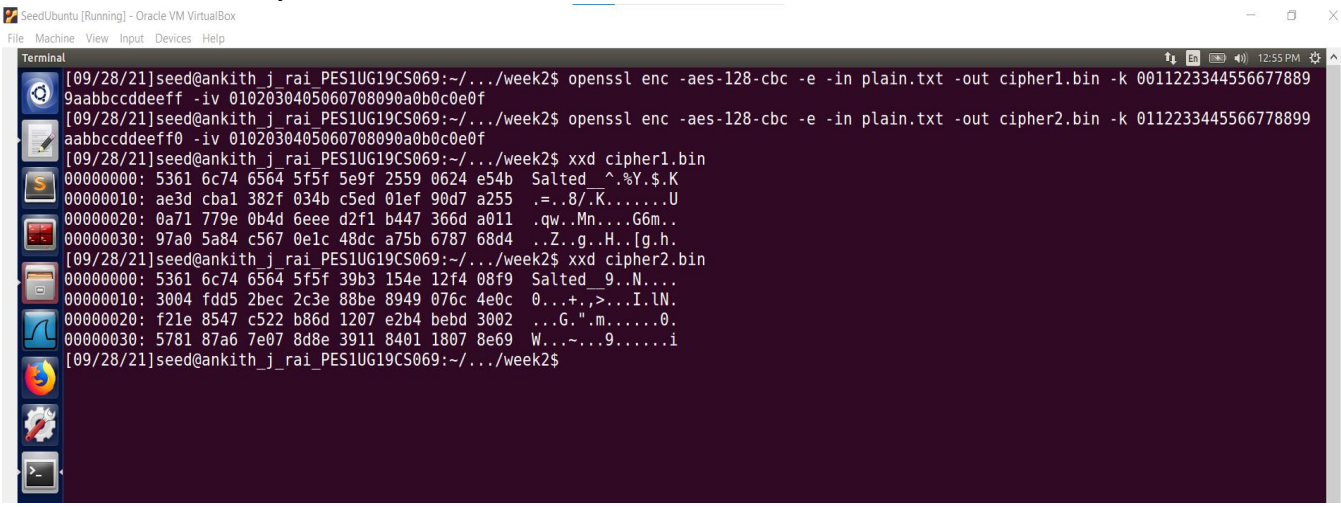


Now we have cipher1.bin which was formed using initial vector **01020304050607080 90a0b0c0e0f** and we have cipher2.bin which was formed using initial vector **102030405060708090a0b0c0d0e0f0.**



We can see that by changing the initial vector by a small amount a completely new cipher text is generated.

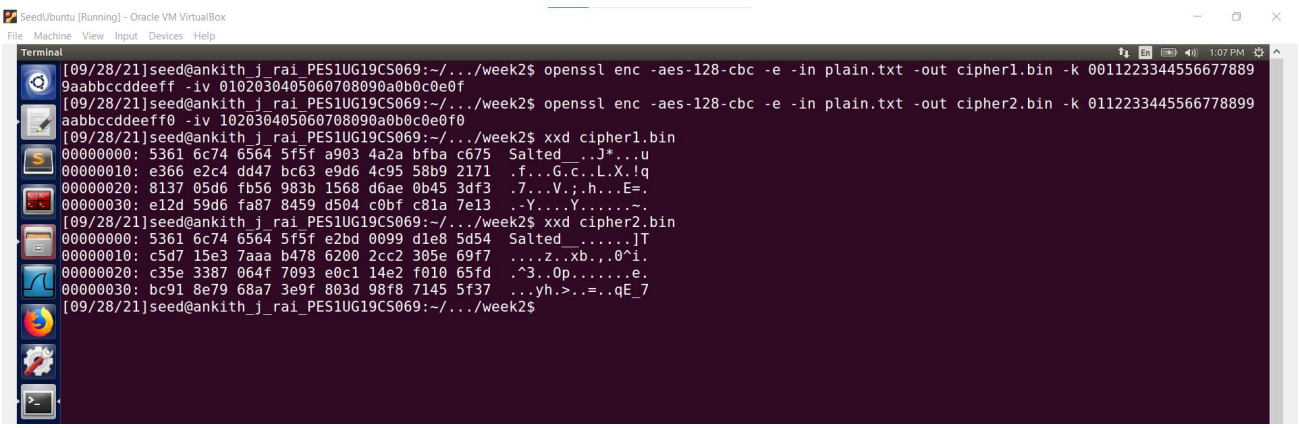
<u>Step 2:</u>Encrypting the same plain text using same initial vector but different keys.



From the above screenshot we can see that cipher1.bin has been formed using key1 **00112233445566778899aabbccddeeff** and we have cipher2.bin has been formed using key2 **01122334455667788 99aabbccddeeff0** .

We can see from the output in cipher1.bin and cipher2.bin that the cipher text in both of them have changed drastically even though there was only a slight change in the keys.

Step 3: Encrypting the same plain text using different initial vector and different keys.



From the above screenshot we can see that cipher1.bin has been formed using (**key1 00112233445566778899aabbccddeeff and initial vector 0102030405060708090a0b0c0e0f)** and we have cipher2.bin has been formed using (**key2 01122334455667788 99aabbccddeeff0 and initial vector 0102030405060708090a0b0c0e0f)** .

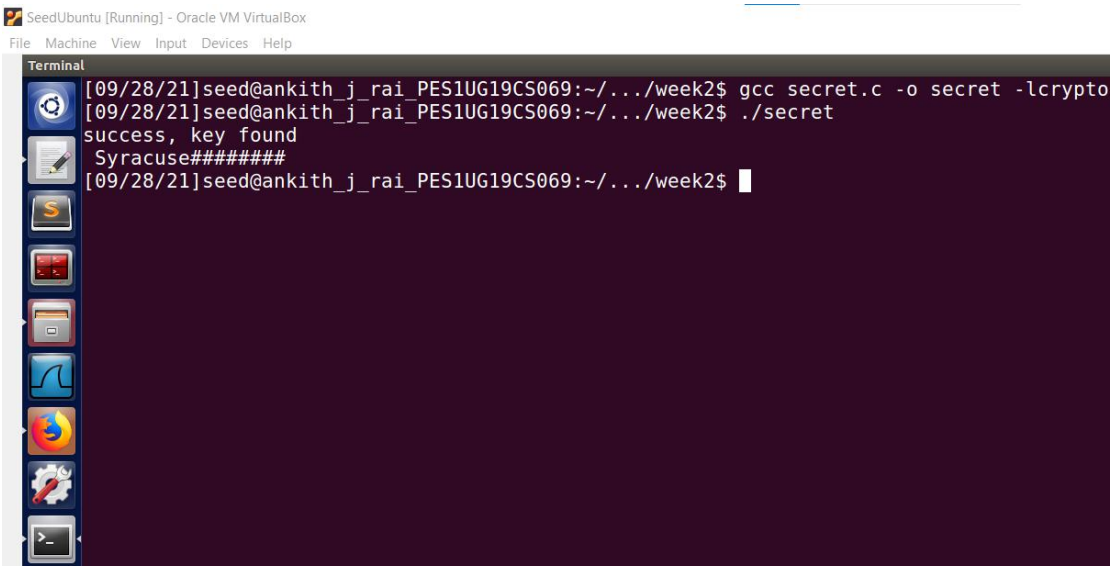
From the above screenshot we can also see that the the cipher text in both of files have changed drastically even though there was only a slight change in the keys and the initial vector.

Task 7: Programming Using the crypto Library

In this task we are going to find the key from a word list given the plain text,initial vector and cypher text.

Here we first create a secret.c file which contains the code to find the key and words.txt which is English words list.

Secret.c code is designed in such a way that the plain text is encrypted using the initial vector and every word in the word.txt(as key).This encrypted text is compared with the cipher text and if there is match of these two the word which is used to encrypt the plain text is returned as key.



On running secret.c we find that the key has been found and the key is **Syracuse########**.The 8 # are present because the key is 16 bytes(I.e 128 bits) in length.