كلية علوم الحاسب والمعلومات
College of Computer and Information Sciences

Secret Key Encryption

Dr. Milad Tezeghdanti

Mohammed Wahaq Alsahli

440015334

CS334 project

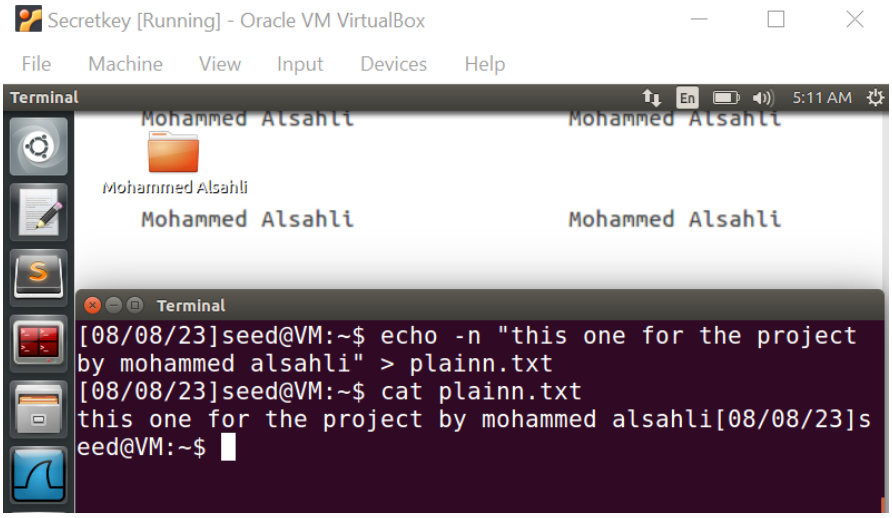
Task #2

1-Using AES-128-CBC:

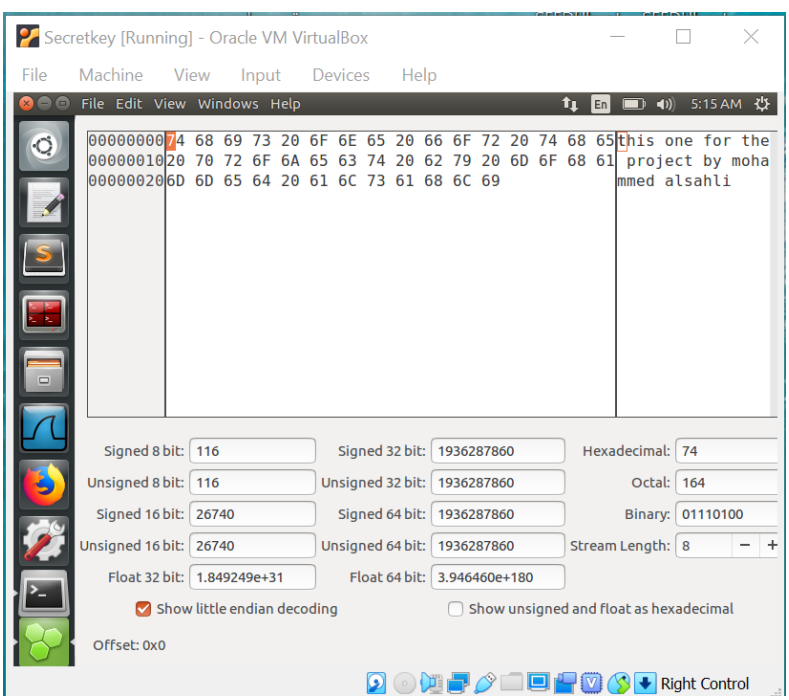
To encrypt we need plaintext, initialization vector, key.

We create the text file:

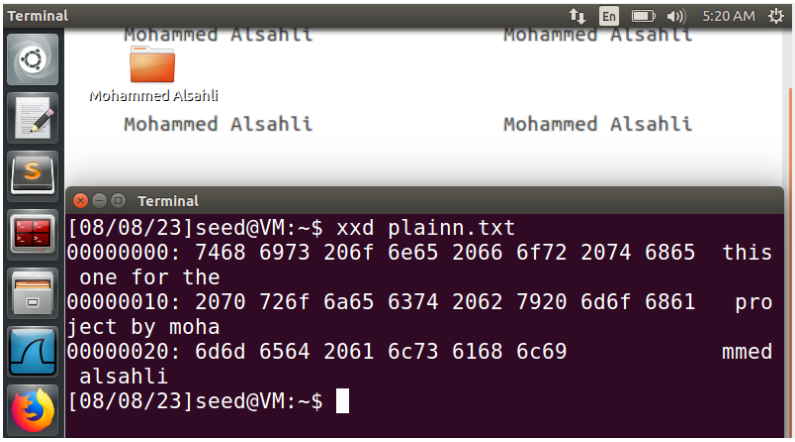
Saved in plainn.txt



The hexadecimal for the plain text:



The size of the text = 44 characters -> “this one for the project by mohammed alsahli”

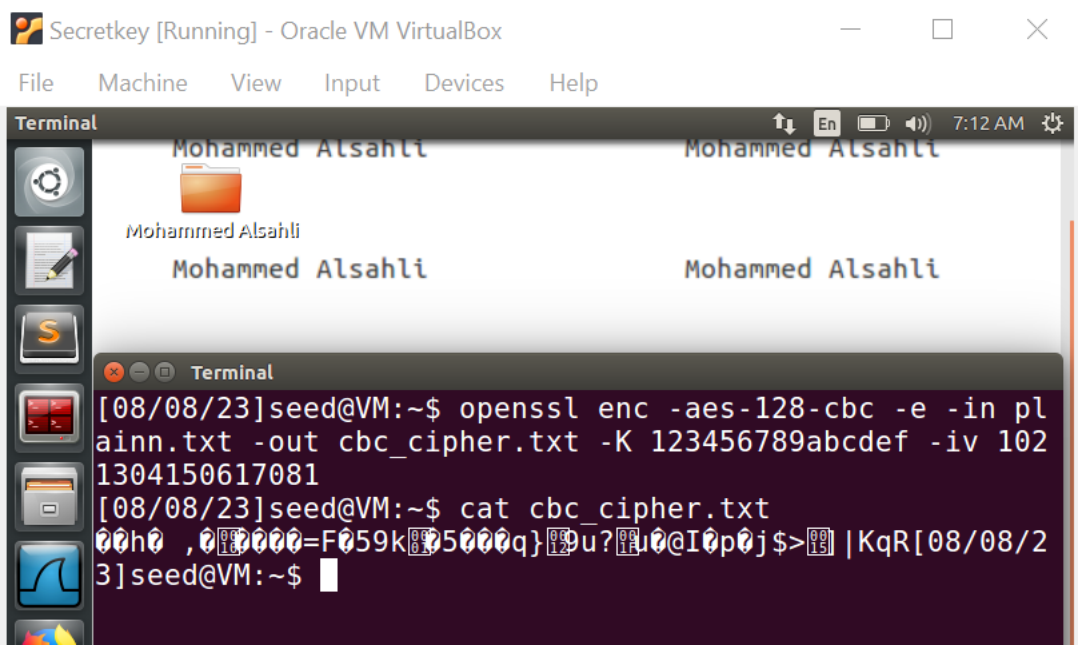


Encrypt using AES-128-CBC:

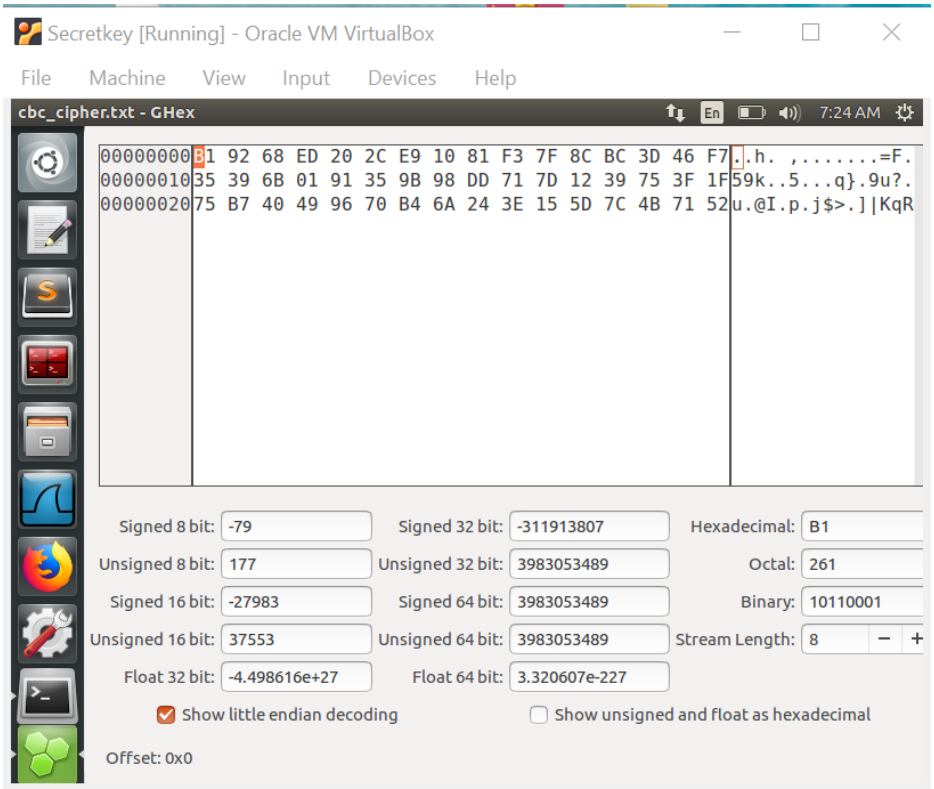
Plain text: “this one for the project by mohammed alsahli”

Key: 123456789abcdef (124-bit)

Iv: 1021304150617081 (124-bit)



The hexadecimal for it:



The observation:

The plaintext is encrypted using AES-128 cipher using (CBC) cipher block chaining mode

The plain text I used contain 44 characters(16-byte(block 1) + 16-byte(block 2) + 12-byte(block 3))

and the ciphered text contains 48 characters (16-byte(block 1) + 16-byte(block 2) + 16-byte(block 3))

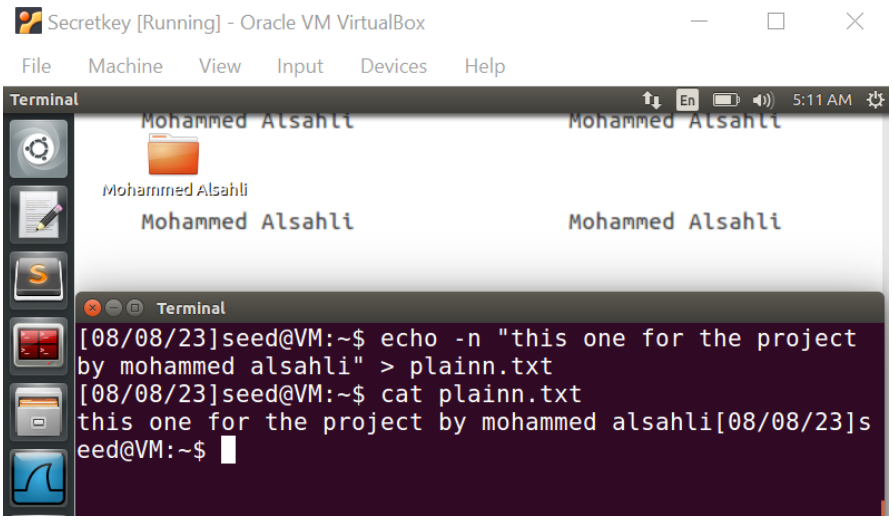
what happened to the third block here is padding filled it so it becomes 16-byte.

2- using DES-EDE-CBC:

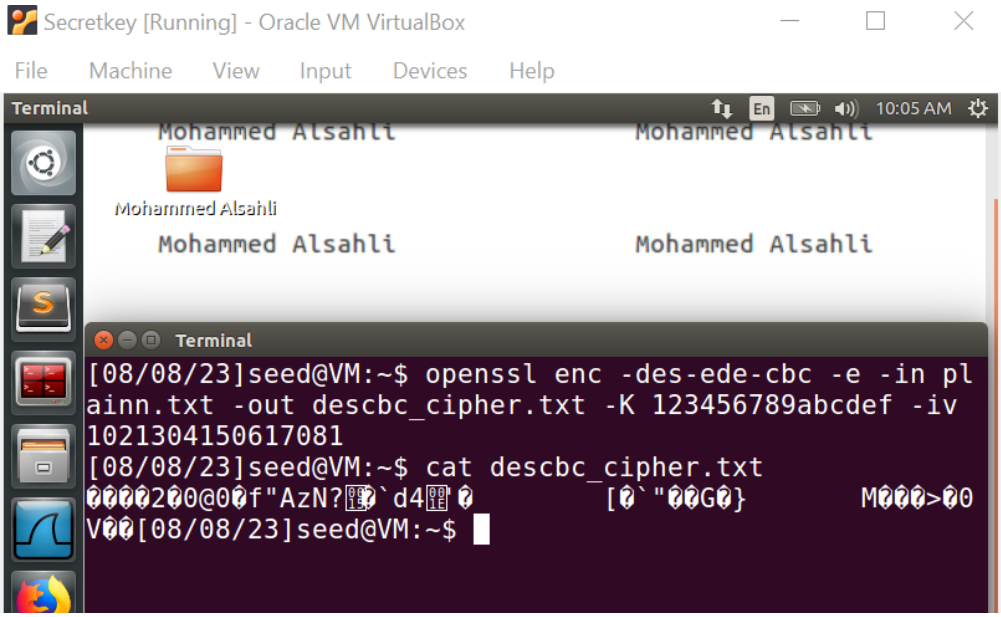
To encrypt we need initialization vector, key.

We create the text file:

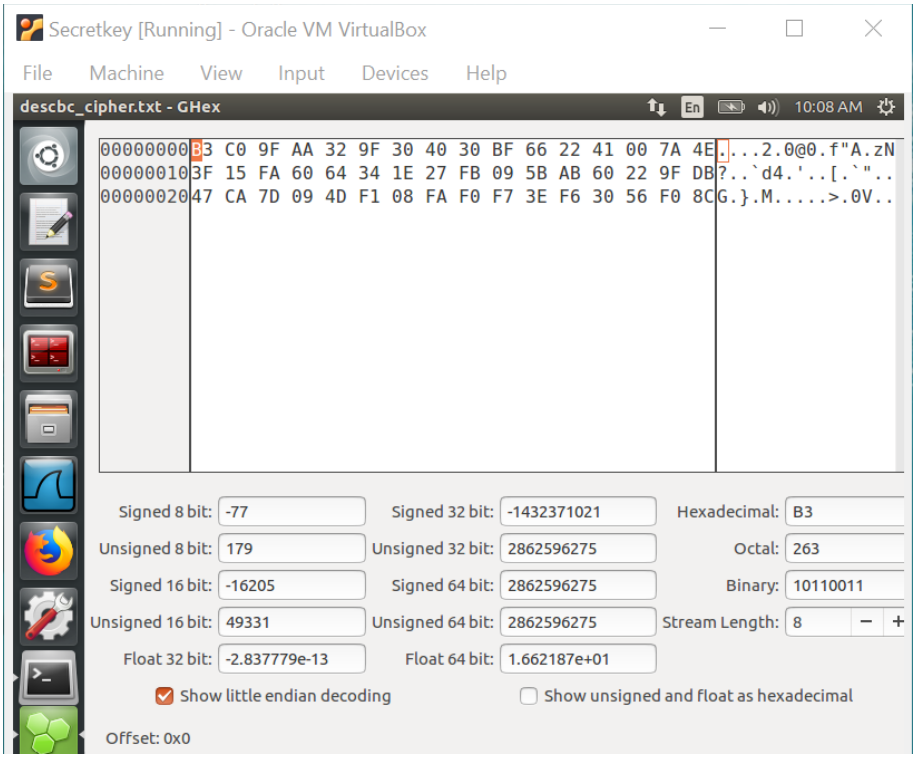
Saved in plainn.txt same as before.



Encryption using DES-EDE-CBC:



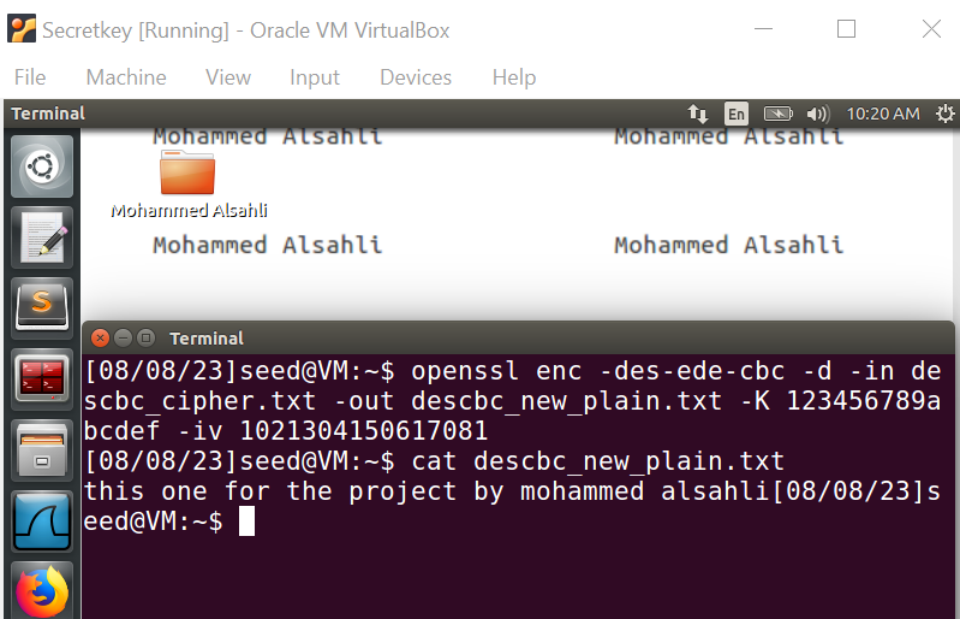
The hexadecimal for it:



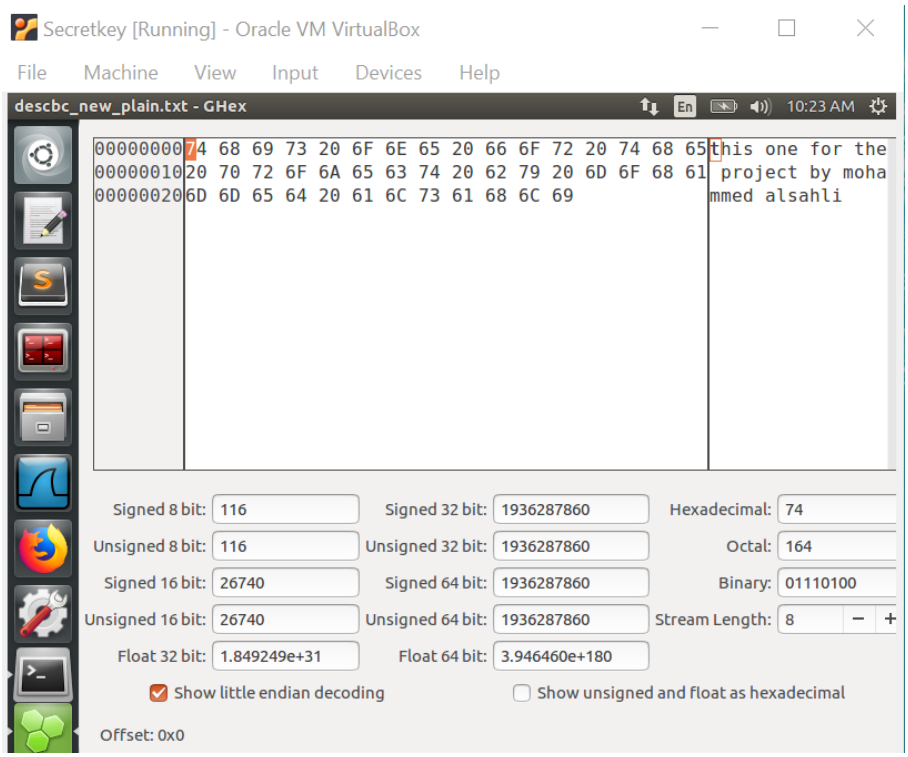
The observation:

DES is a block cipher and encrypts data in blocks of size of 64 bits each, which means 64 bits of plain text go as the input to DES The same algorithm and key are used for encryption and decryption, with minor differences

The decryption:



The hexadecimal for it:

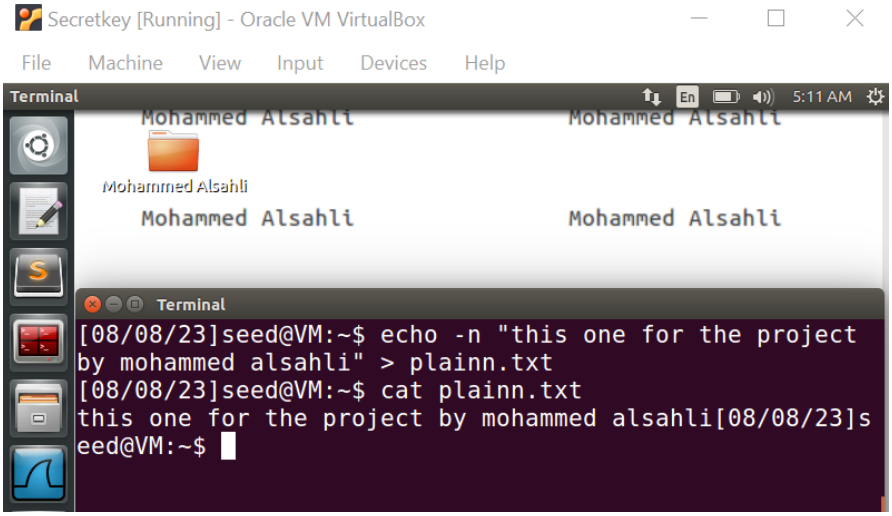


3- using AES-128-CFB:

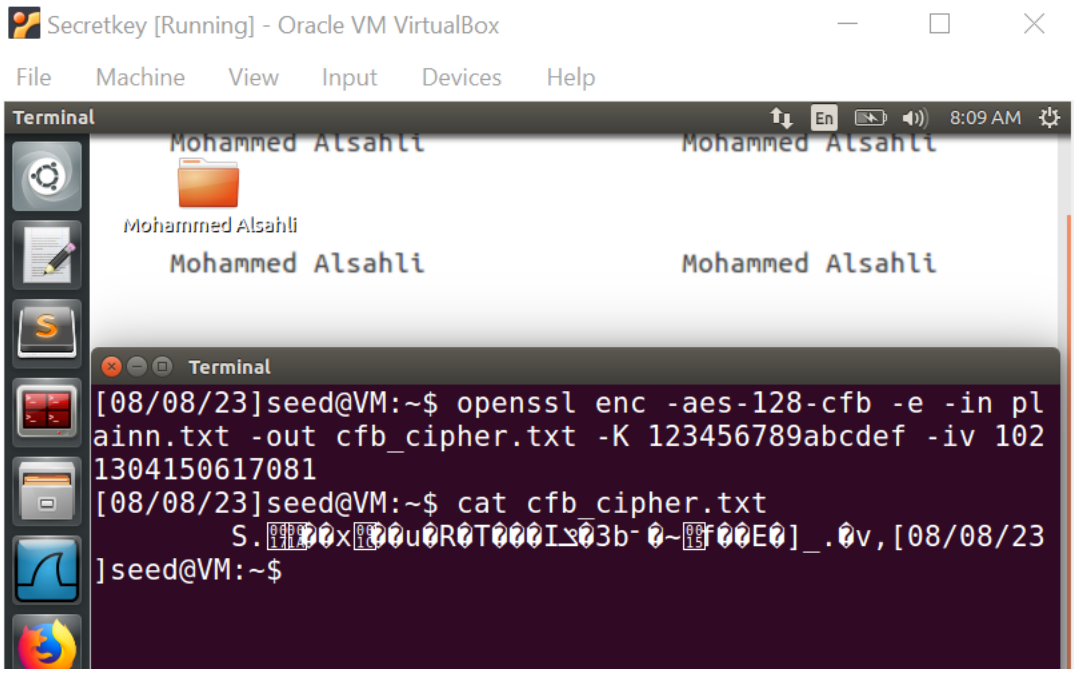
To encrypt we need initialization vector, key.

We create the text file:

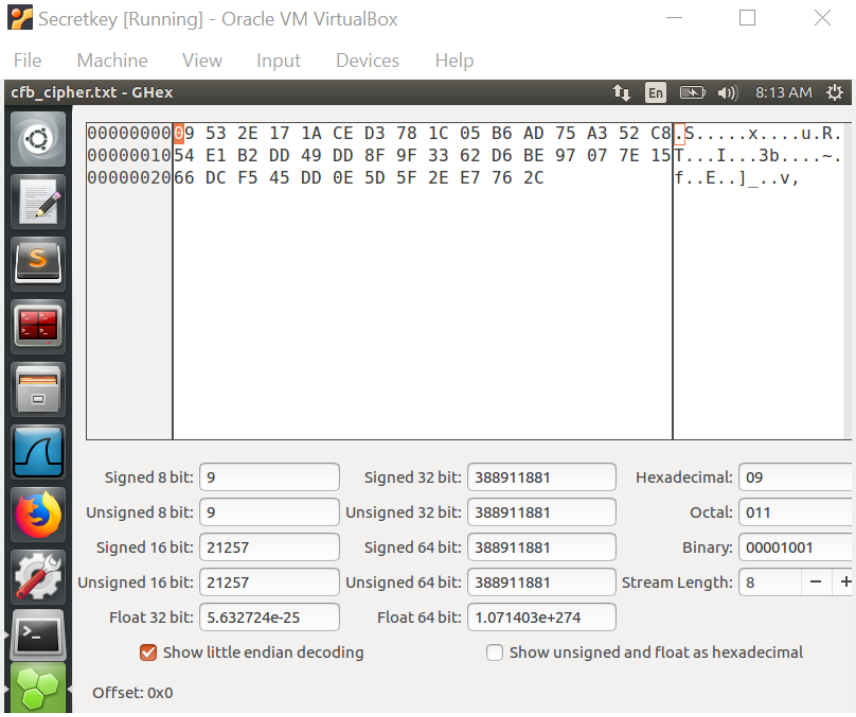
Saved in plainn.txt same as before.



Encryption using AES-128-CFB:



the hexadecimal for it:



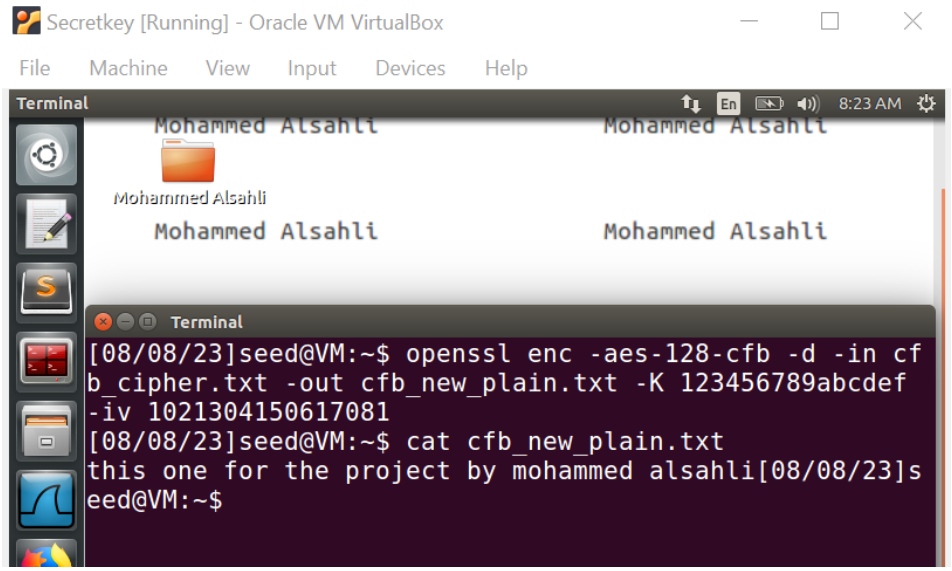
The observation:

the plaintext contains 44 characters and the ciphertext contains 44 characters Because it's a stream it's deals with every byte individually rather than the block method where it had to do padding

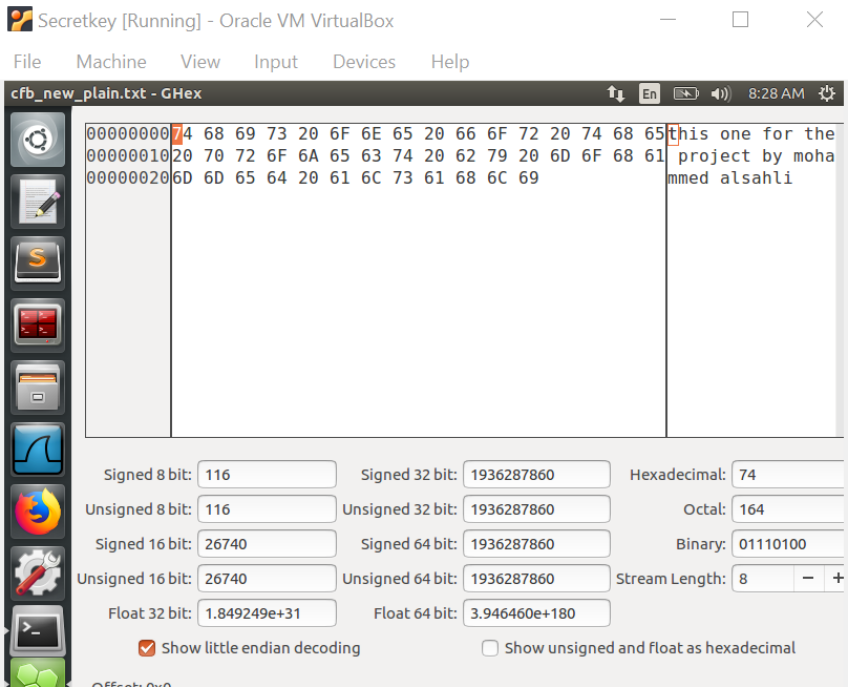
Decryption using AES-128-CFB:

Key: 123456789abcdef (124-bit)

Iv: 1021304150617081 (124-bit)



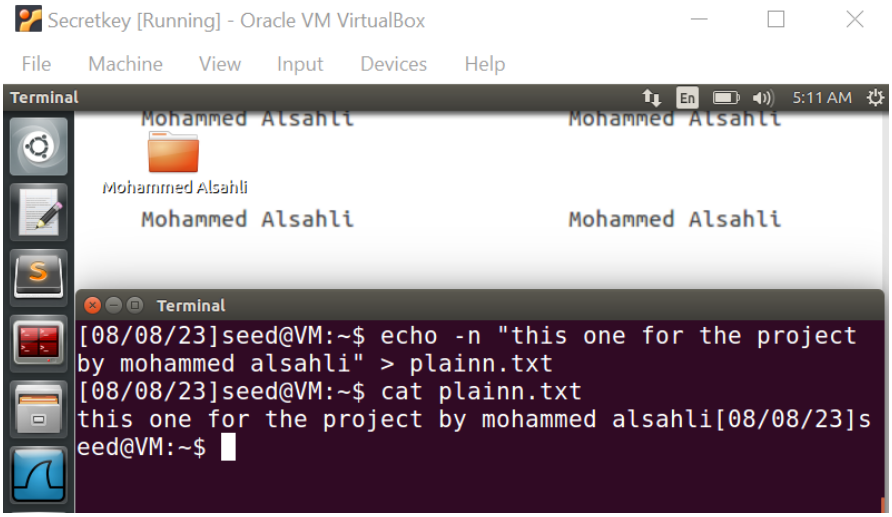
The hexadecimal for it:



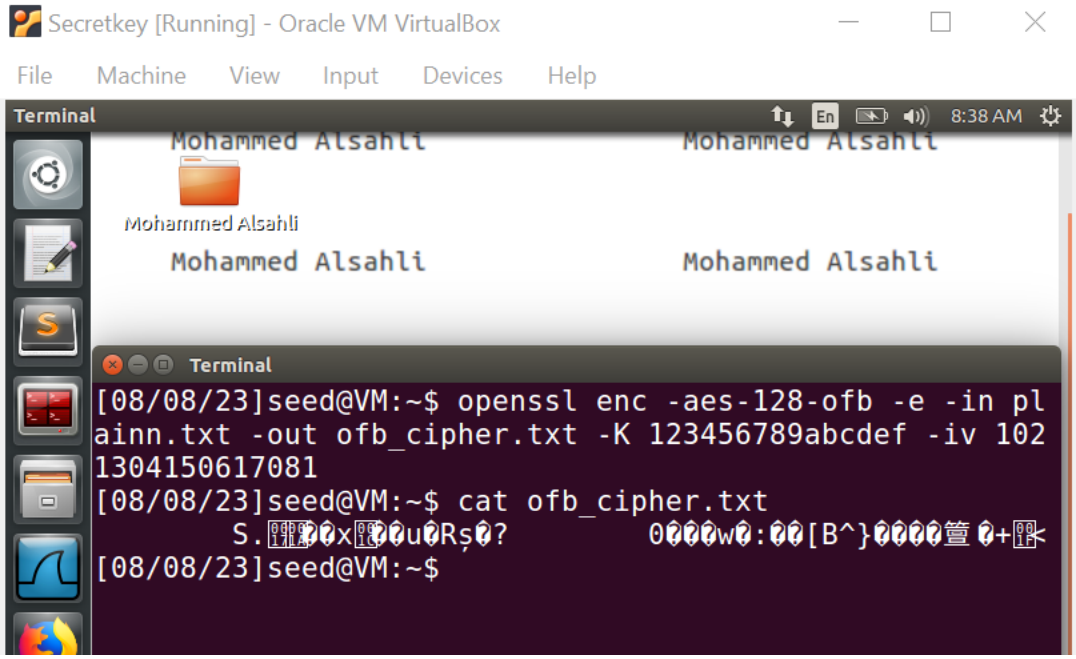
4- using AES-128-OFB:

We create the text file:

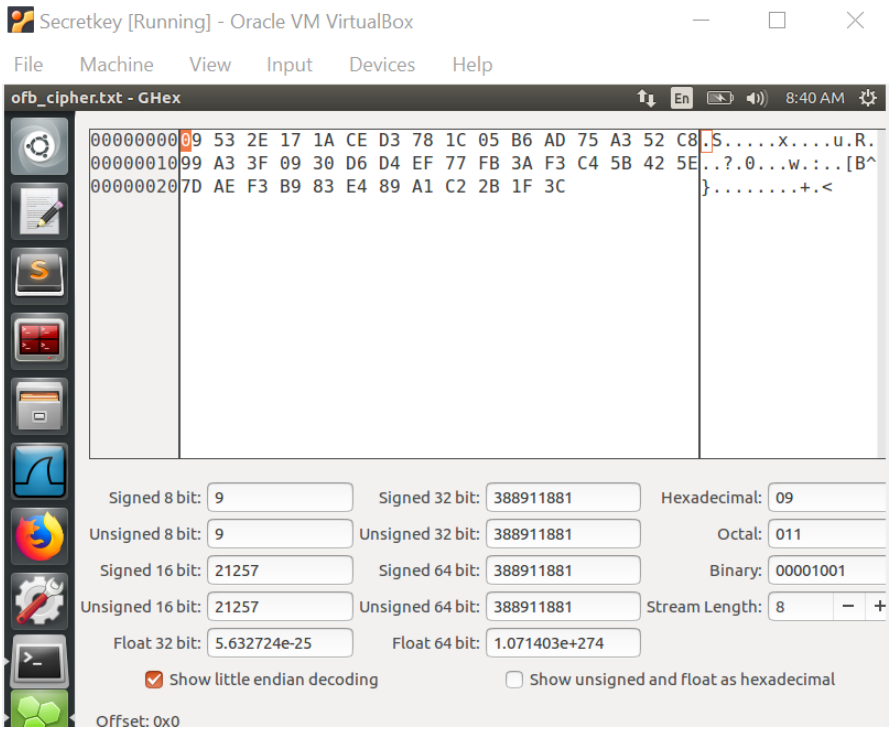
Saved in plainn.txt same as before.



Encrypt using AES-128-OFB:



The hexadecimal for it:



The observation:

The OFB works parallelly and similar to CFB

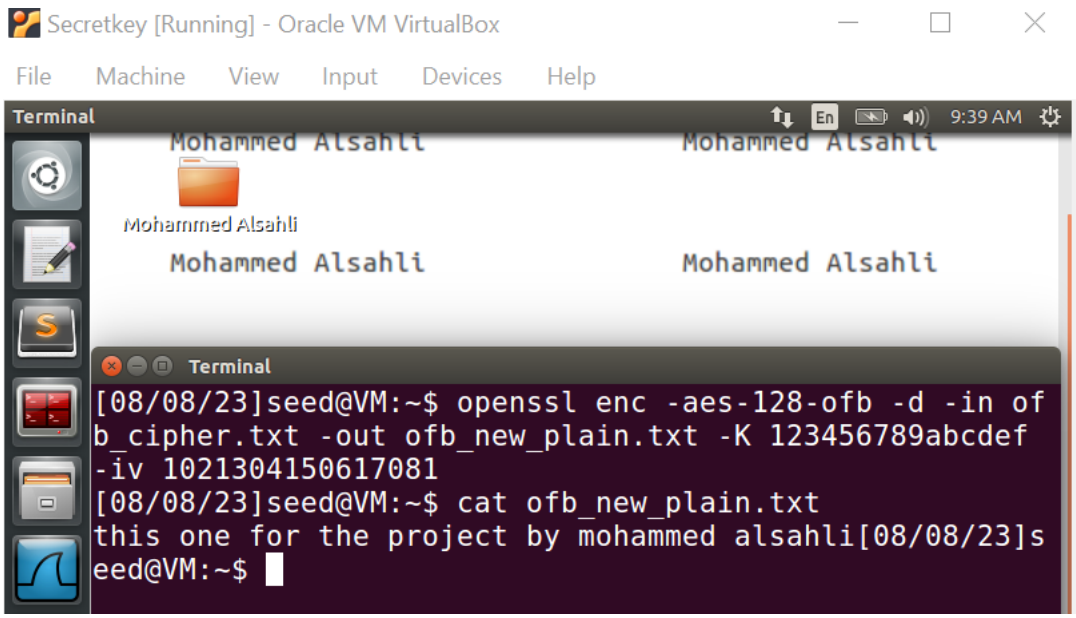
the plaintext contains 44 characters and the ciphertext contains 44 characters

Because OFB is a stream cipher and the plain text XOR with the ciphered text

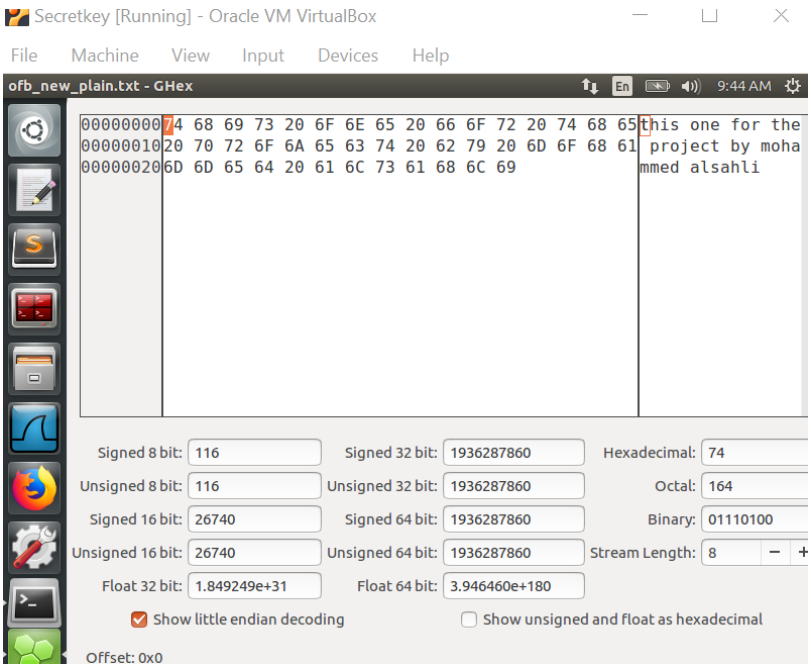
Decryption of AES-128-OFB:

Key: 123456789abcdef (124-bit)

Iv: 1021304150617081 (124-bit)

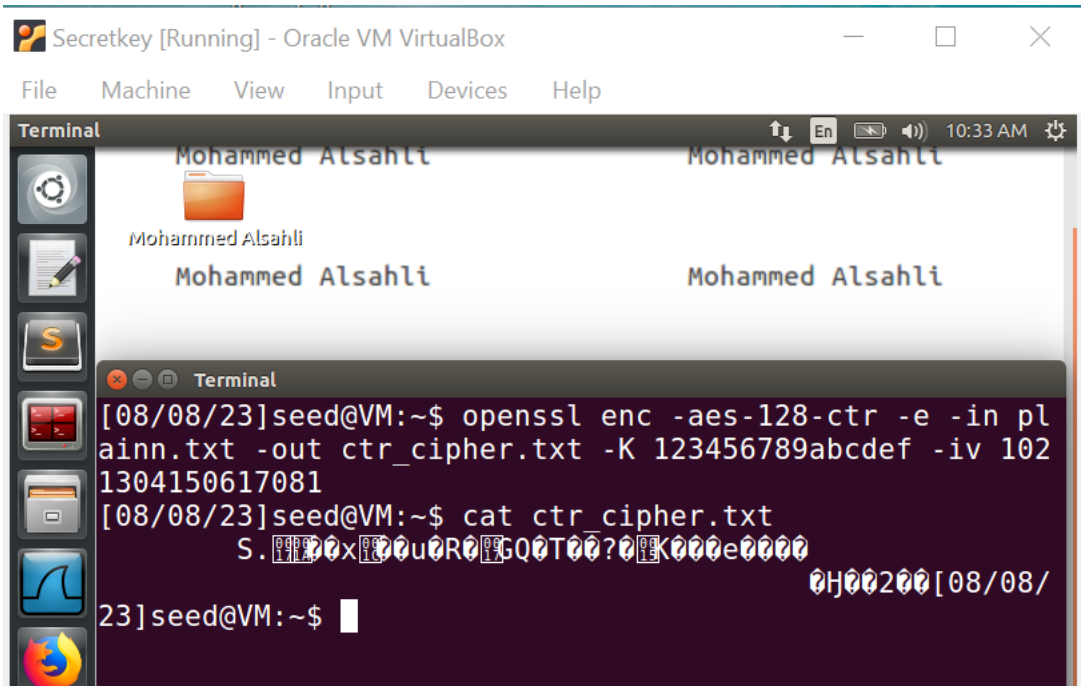


The hexadecimal for it:

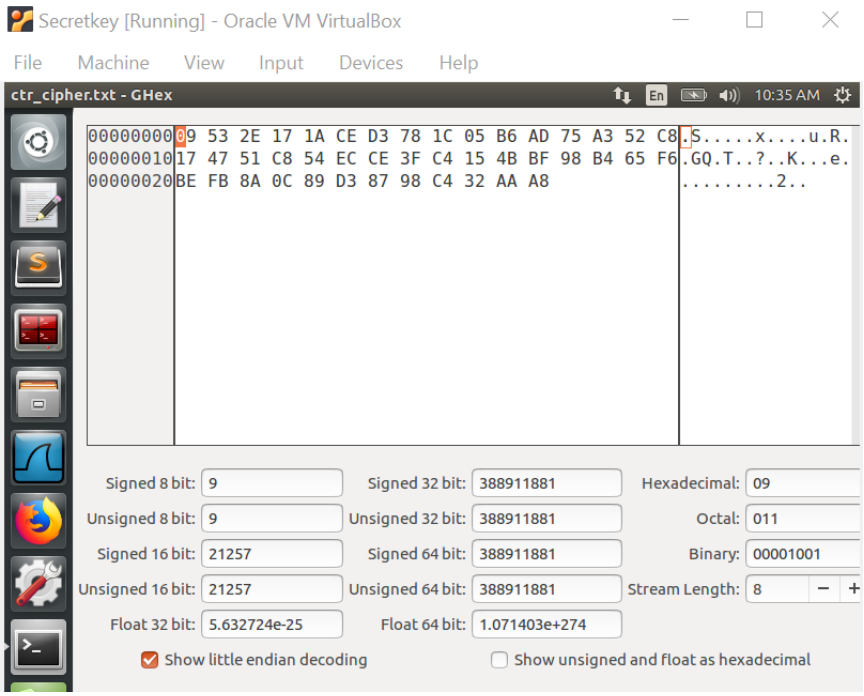


5- AES-128-CTR:

Encrypt using AES-128-CTR:



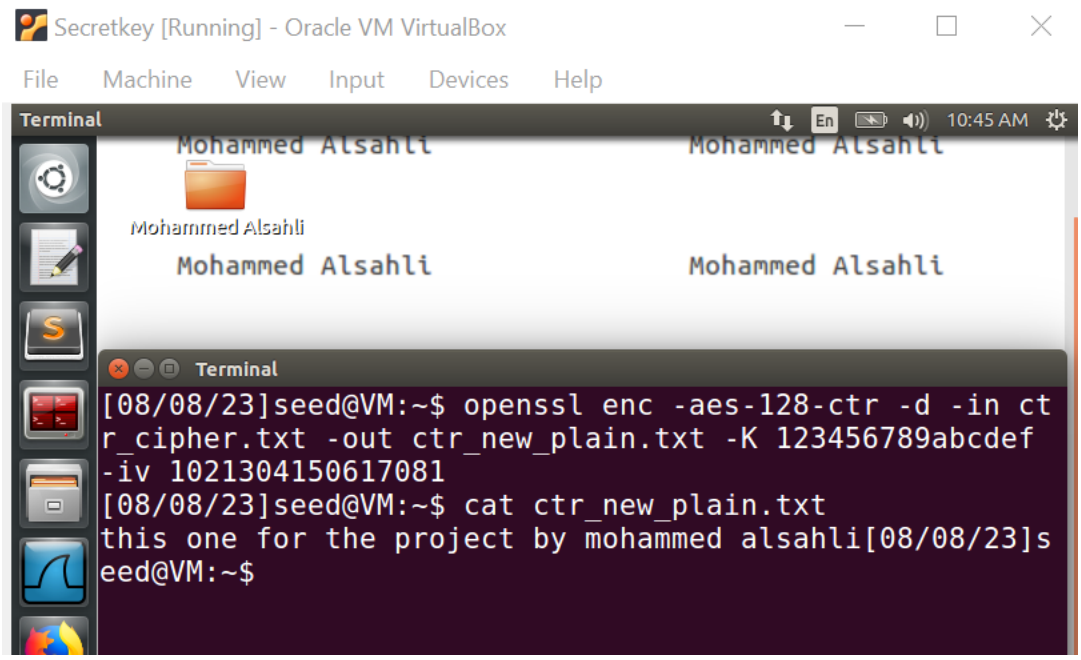
The hexadecimal for it:



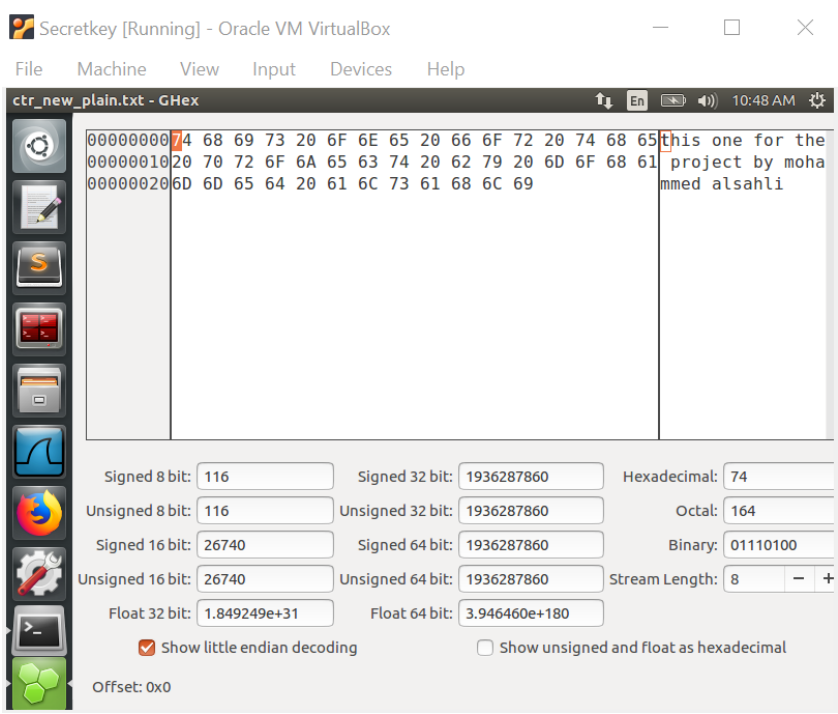
The observation:

mode is a typical block cipher mode of operation using block cipher algorithm. It is parallel. CTR is similar to OFB as it also involves XOR-ing a sequence of pad vectors with the plaintext and ciphertext blocks

The decryption of AES-128-CTR:

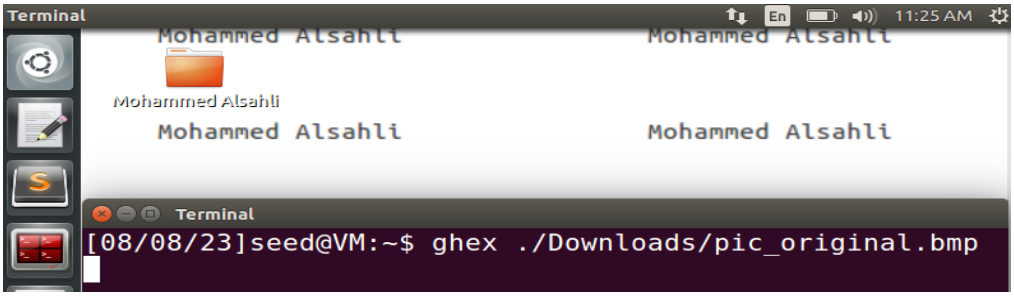


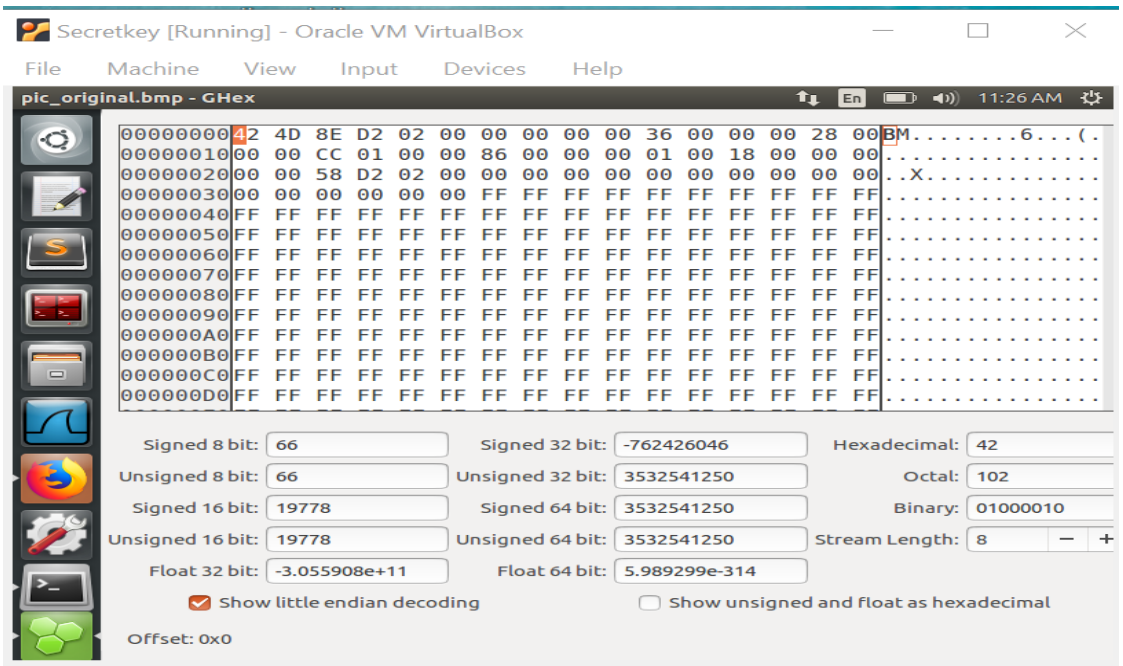
The hexadecimal for it:



Task #3:

The first 3 rows in the hexadecimal is the header (52-byte)

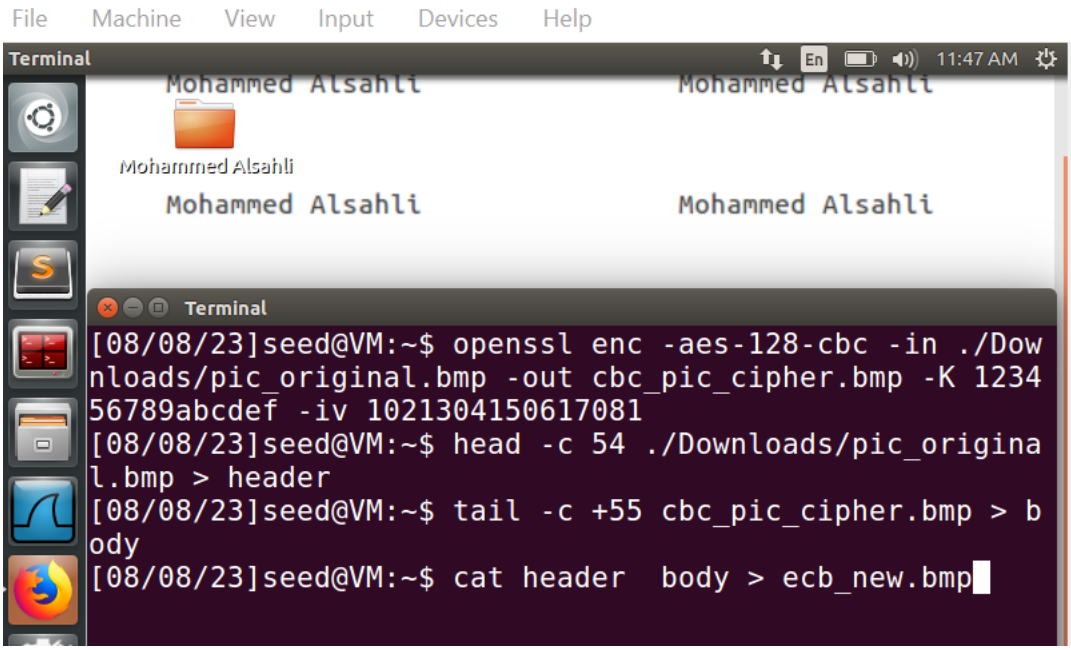




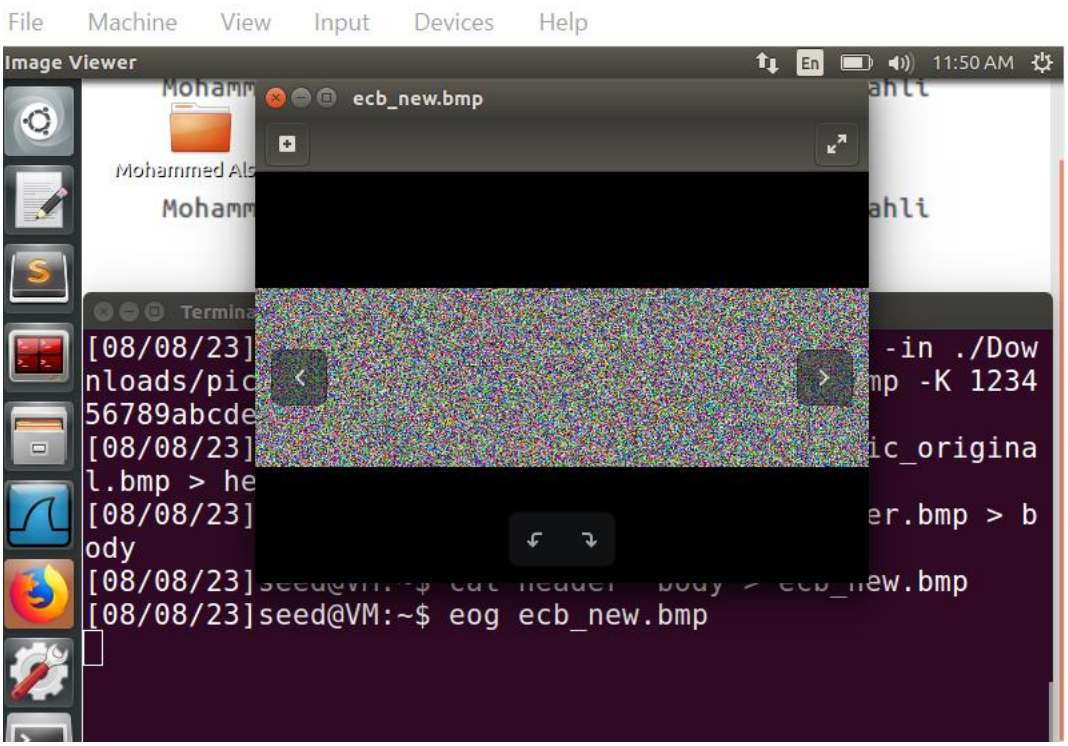
Key: 123456789abcdef (124-bit)

Iv: 1021304150617081 (124-bit)

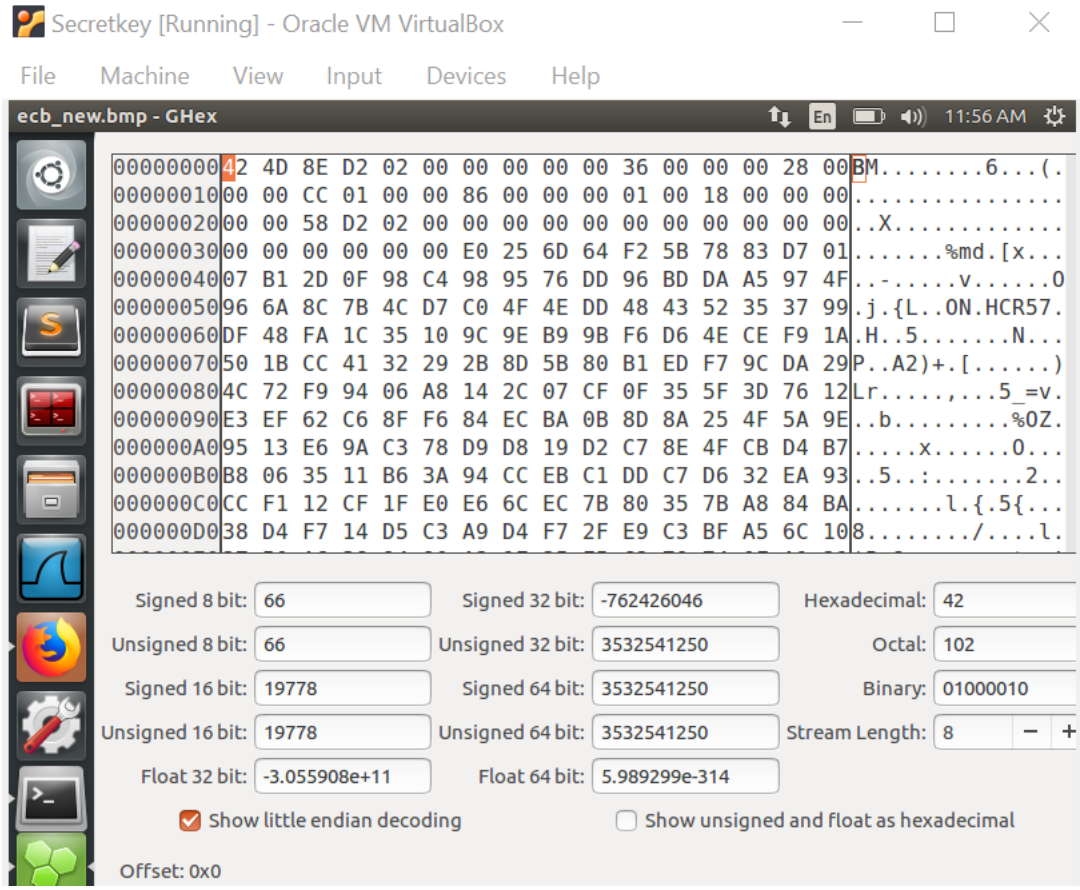
Encrypting the image using AES-128-CBC:



We used the command eog



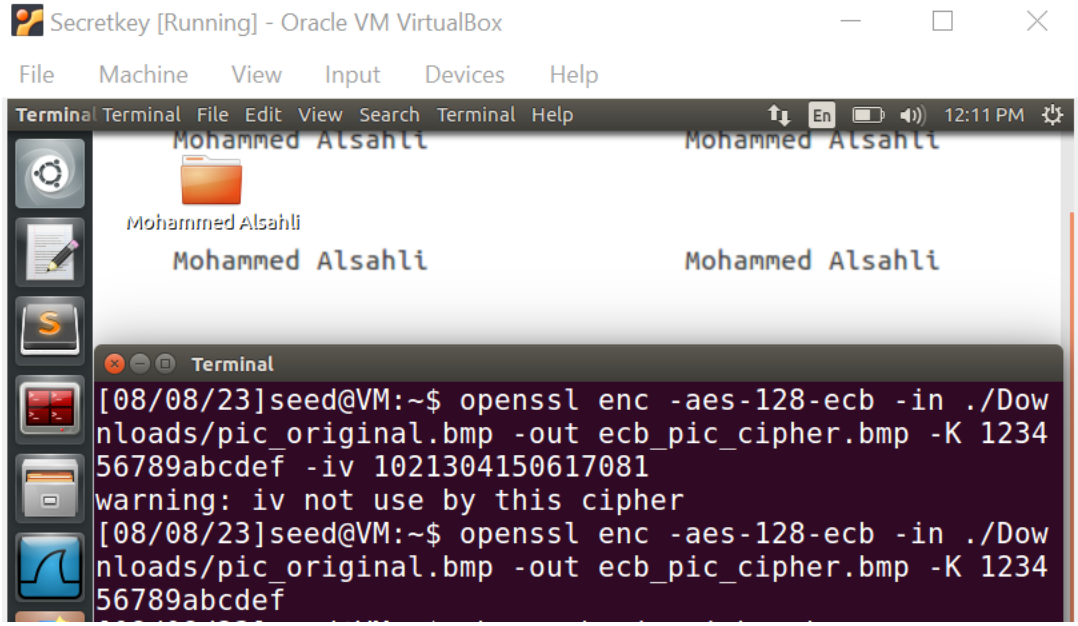
The hexadecimal for it:



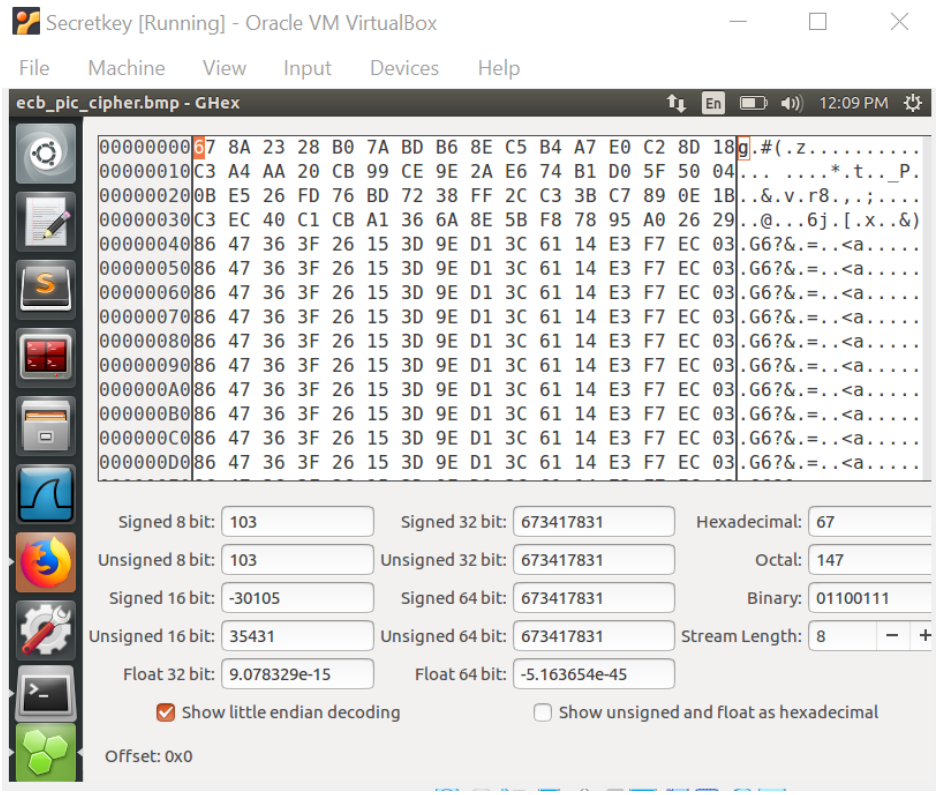
Observation:

The first 54 bits is replaced with the 54 bits of the original file Which makes it visible but, any information about the original image is not visible in this file because the CBC mode generates cyphertext for repeating plaintext

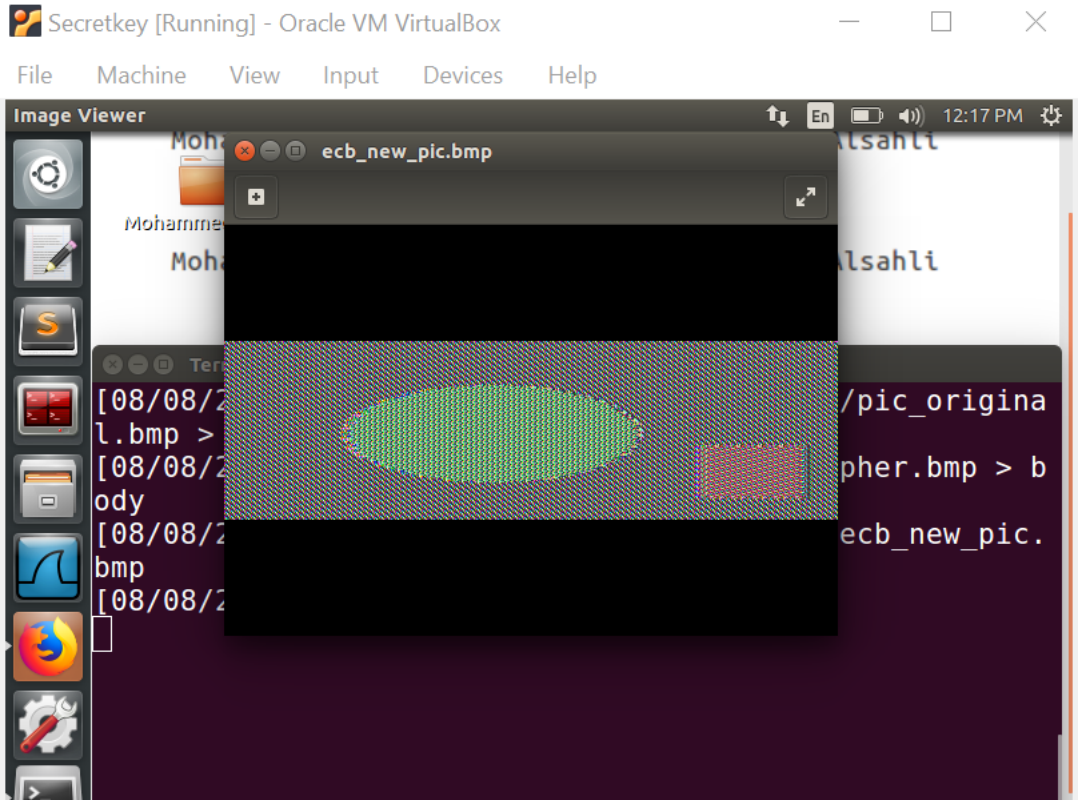
Encrypting using AES-128-ECB:



The hexadecimal for it:



view the image by eog



The observation:

The first 54 bits is replaced with the 54 bits of the original file Which makes it visible but, because for every block cipher encryption the output will be the same ciphertext , we notice that the content of the image is visible but it's still not the same as the original

Task #7

Using AES-128-CBC:

Plain text: this is top secret.

Key: From the dictionary :

ahead########### -> 6168656164232323232323232323232323 (to hex)

iv: aabbccddeeff00998877665544332211 (128-bit)

Cipher text:
764aa26b55a4da654df6b19e4bce00f4ed05e09346fb0e762583cb7da2ac93a2

Run:

```
Key search processing done.. >>>>
explain test is: this is top secret.
Key used in ASKII: ahead###########
key used in hex: 6168656164232323232323232323232323
Encrypted text in hex: 764aa26b55a4da654df6b19e4bce00f4ed05e09346fb0e762583cb7da2ac93a2
```