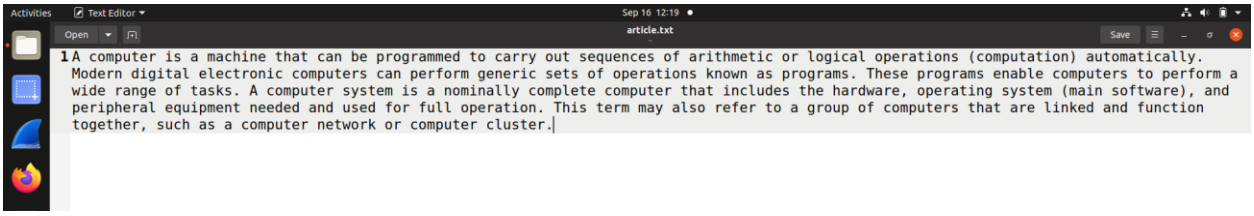
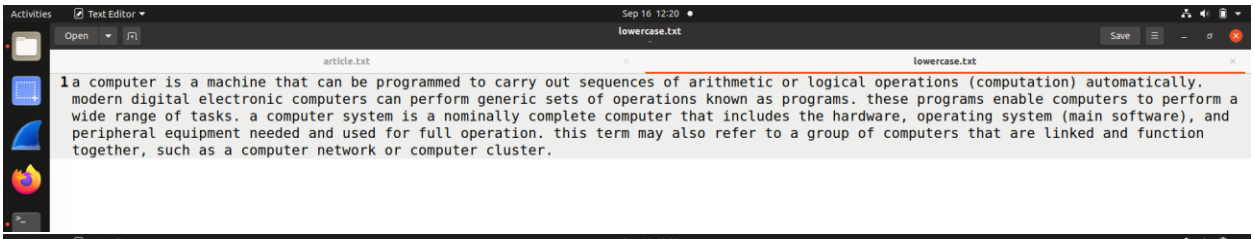
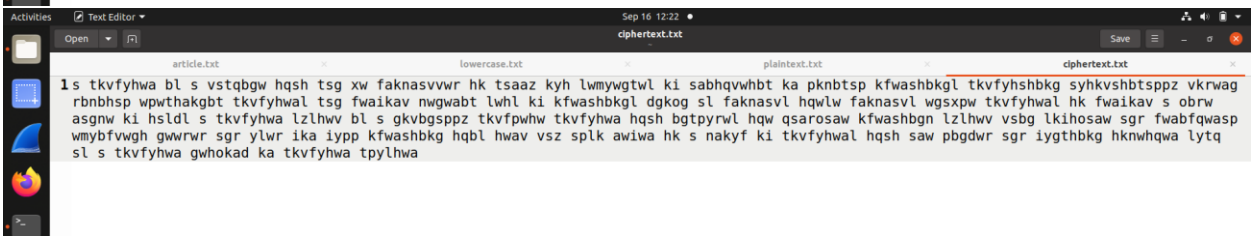
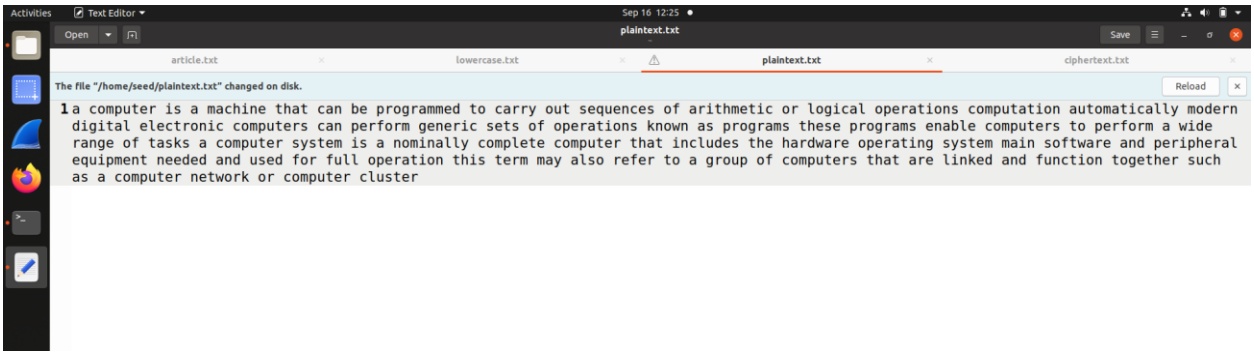
SEED LABS – ENCRYPTION AND DECRYPTION

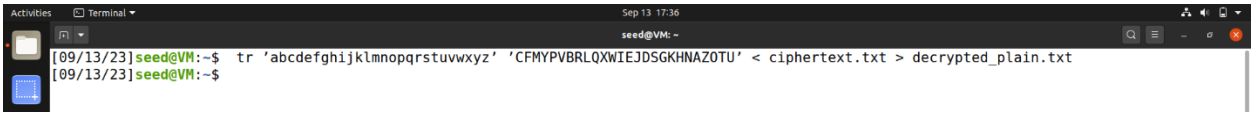
Frequency Analysis





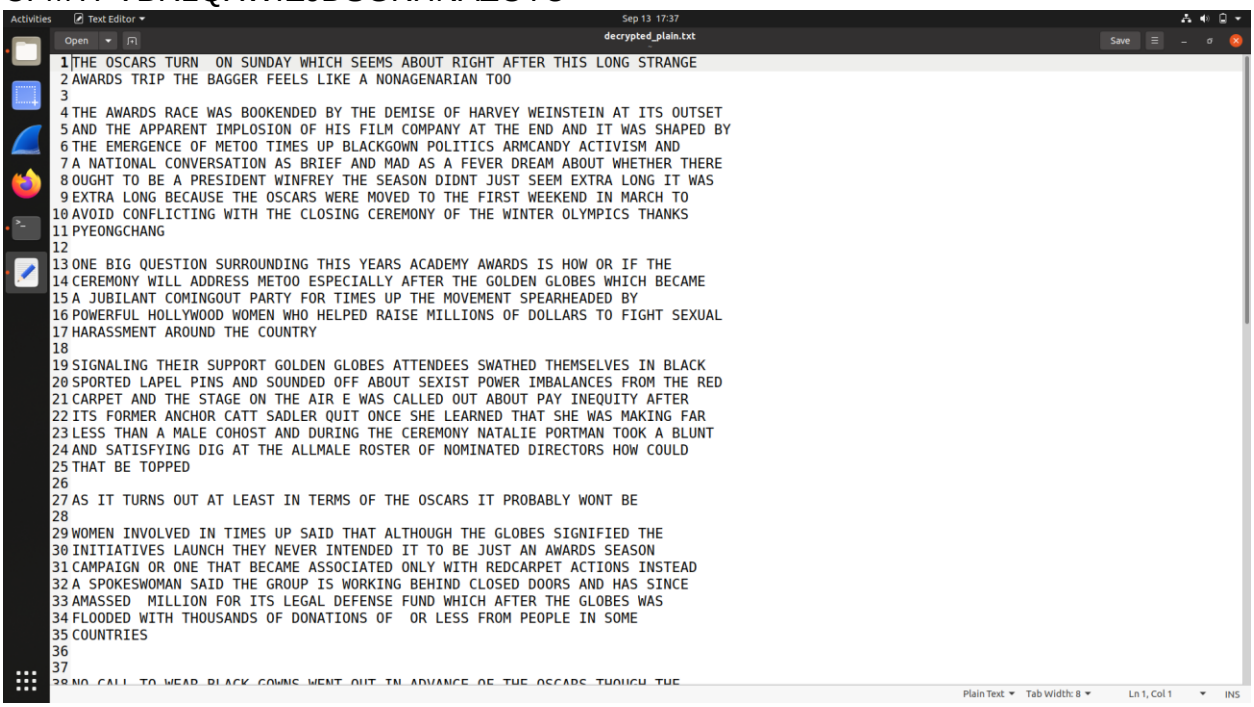


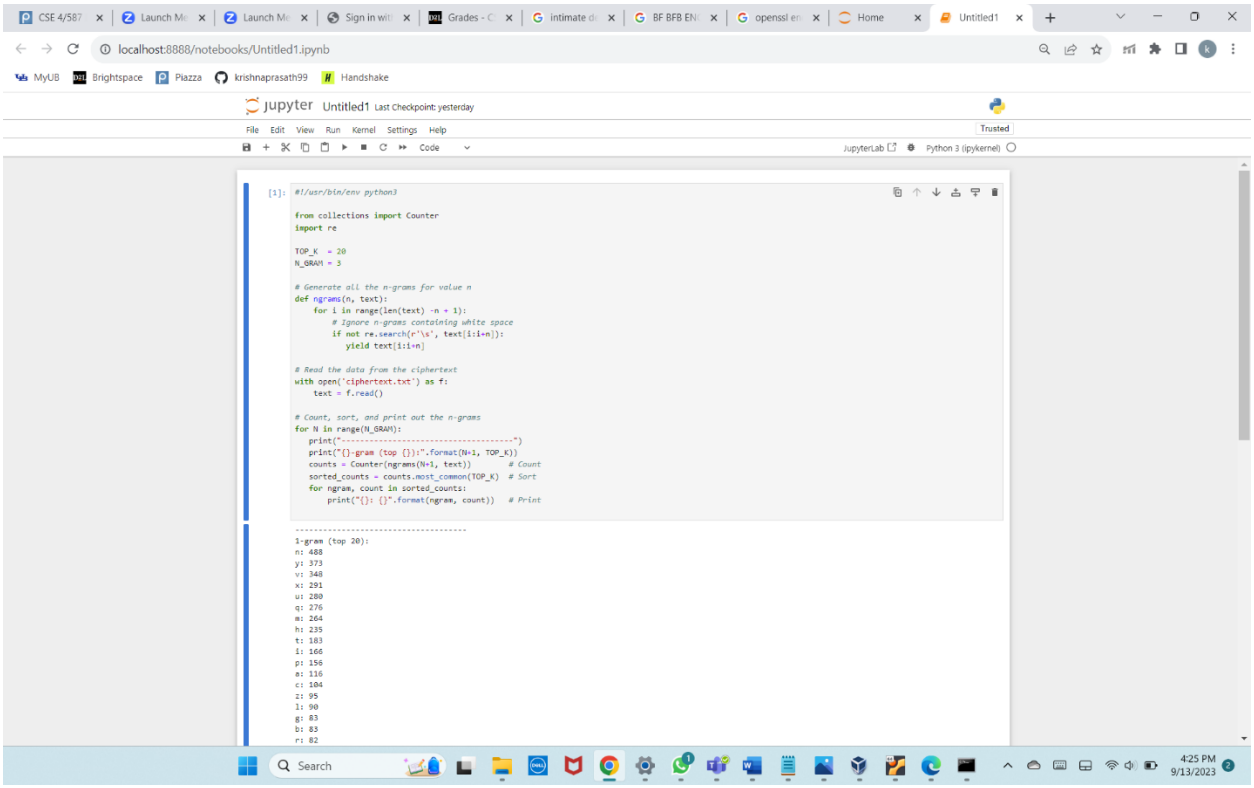




ENCRYPTION KEY :

CFMYPVBRLQXWIEJDSGKHNAZOTU





TASK DONE AND OBSERVATION:

- By running the given code snippet I was able to get the words with highest frequencies.
- Using the frequency analysis method, I have referred the words with higher frequency (3 gram, 2 gram, 1 gram frequency) in the wikipedia and replaced the encrypted data it with the corresponding plain text.
- Did this multiple times using hit and trial method manually for all the alphabets.
- Finally found the correct encryption key and decrypted and displayed the corresponding plaintext.
- CFMYPVBRLQXWIEJDSGKHNAZOTU is the Encryption key
- Decrypted data is printed below for reference.

DECRYPTED PLAIN TEXT :

THE OSCARS TURN ON SUNDAY WHICH SEEMS ABOUT RIGHT AFTER THIS LONG STRANGE
AWARDS TRIP THE BAGGER FEELS LIKE A NONAGENARIAN TOO

THE AWARDS RACE WAS BOOKENDED BY THE DEMISE OF HARVEY WEINSTEIN AT ITS OUTSET
AND THE APPARENT IMPLOSION OF HIS FILM COMPANY AT THE END AND IT WAS SHAPED BY
THE EMERGENCE OF METOO TIMES UP BLACKGOWN POLITICS ARMCANDY ACTIVISM AND
A NATIONAL CONVERSATION AS BRIEF AND MAD AS A FEVER DREAM ABOUT WHETHER THERE
OUGHT TO BE A PRESIDENT WINFREY THE SEASON DIDNT JUST SEEM EXTRA LONG IT WAS
EXTRA LONG BECAUSE THE OSCARS WERE MOVED TO THE FIRST WEEKEND IN MARCH TO
AVOID CONFLICTING WITH THE CLOSING CEREMONY OF THE WINTER OLYMPICS THANKS
PYEONGCHANG

ONE BIG QUESTION SURROUNDING THIS YEARS ACADEMY AWARDS IS HOW OR IF THE CEREMONY WILL ADDRESS METOO ESPECIALLY AFTER THE GOLDEN GLOBES WHICH BECAME
A JUBILANT COMINGOUT PARTY FOR TIMES UP THE MOVEMENT SPEARHEADED BY
POWERFUL HOLLYWOOD WOMEN WHO HELPED RAISE MILLIONS OF DOLLARS TO FIGHT SEXUAL
HARASSMENT AROUND THE COUNTRY

SIGNALING THEIR SUPPORT GOLDEN GLOBES ATTENDEES SWATHED THEMSELVES IN BLACK
SPORTED LAPEL PINS AND SOUNDED OFF ABOUT SEXIST POWER IMBALANCES FROM THE RED
CARPET AND THE STAGE ON THE AIR E WAS CALLED OUT ABOUT PAY INEQUITY AFTER ITS FORMER ANCHOR CATT SADLER QUIT ONCE SHE LEARNED THAT SHE WAS MAKING FAR
LESS THAN A MALE COHOST AND DURING THE CEREMONY NATALIE PORTMAN TOOK A BLUNT
AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOMINATED DIRECTORS HOW COULD
THAT BE TOPPED

AS IT TURNS OUT AT LEAST IN TERMS OF THE OSCARS IT PROBABLY WONT BE

WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SIGNIFIED THE INITIATIVES LAUNCH THEY NEVER INTENDED IT TO BE JUST AN AWARDS SEASON CAMPAIGN OR ONE THAT BECAME ASSOCIATED ONLY WITH REDCARPET ACTIONS INSTEAD
A SPOKESWOMAN SAID THE GROUP IS WORKING BEHIND CLOSED DOORS AND HAS SINCE
AMASSED MILLION FOR ITS LEGAL DEFENSE FUND WHICH AFTER THE GLOBES WAS FLOODED WITH THOUSANDS OF DONATIONS OF OR LESS FROM PEOPLE IN SOME COUNTRIES

NO CALL TO WEAR BLACK GOWNS WENT OUT IN ADVANCE OF THE OSCARS THOUGH THE
MOVEMENT WILL ALMOST CERTAINLY BE REFERENCED BEFORE AND DURING THE CEREMONY
ESPECIALLY SINCE VOCAL METOO SUPPORTERS LIKE ASHLEY JUDD LAURA DERN AND NICOLE KIDMAN ARE SCHEDULED PRESENTERS

ANOTHER FEATURE OF THIS SEASON NO ONE REALLY KNOWS WHO IS GOING TO WIN BEST
PICTURE ARGUABLY THIS HAPPENS A LOT OF THE TIME INARGUABLY THE NAILBITER NARRATIVE ONLY SERVES THE AWARDS HYPE MACHINE BUT OFTEN THE PEOPLE FORECASTING
THE RACE SOCALLED OSCAROLOGISTS CAN MAKE ONLY EDUCATED GUESSES

THE WAY THE ACADEMY TABULATES THE BIG WINNER DOESNT HELP IN EVERY OTHER CATEGORY THE NOMINEE WITH THE MOST VOTES WINS BUT IN THE BEST PICTURE CATEGORY VOTERS ARE ASKED TO LIST THEIR TOP MOVIES IN PREFERENTIAL ORDER IF A
MOVIE GETS MORE THAN PERCENT OF THE FIRSTPLACE VOTES IT WINS WHEN NO MOVIE MANAGES THAT THE ONE WITH THE FEWEST FIRSTPLACE VOTES IS ELIMINATED AND

ITS VOTES ARE REDISTRIBUTED TO THE MOVIES THAT GARNERED THE ELIMINATED BALLOTS
SECONDPLACE VOTES AND THIS CONTINUES UNTIL A WINNER EMERGES

IT IS ALL TERRIBLY CONFUSING BUT APPARENTLY THE CONSENSUS FAVORITE COMES OUT
AHEAD IN THE END THIS MEANS THAT ENDOFSEASON AWARDS CHATTER INVARIABLY INVOLVES TORTURED SPECULATION ABOUT WHICH FILM WOULD MOST LIKELY BE VOTERS
SECOND OR THIRD FAVORITE AND THEN EQUALLY TORTURED CONCLUSIONS ABOUT WHICH
FILM MIGHT PREVAIL

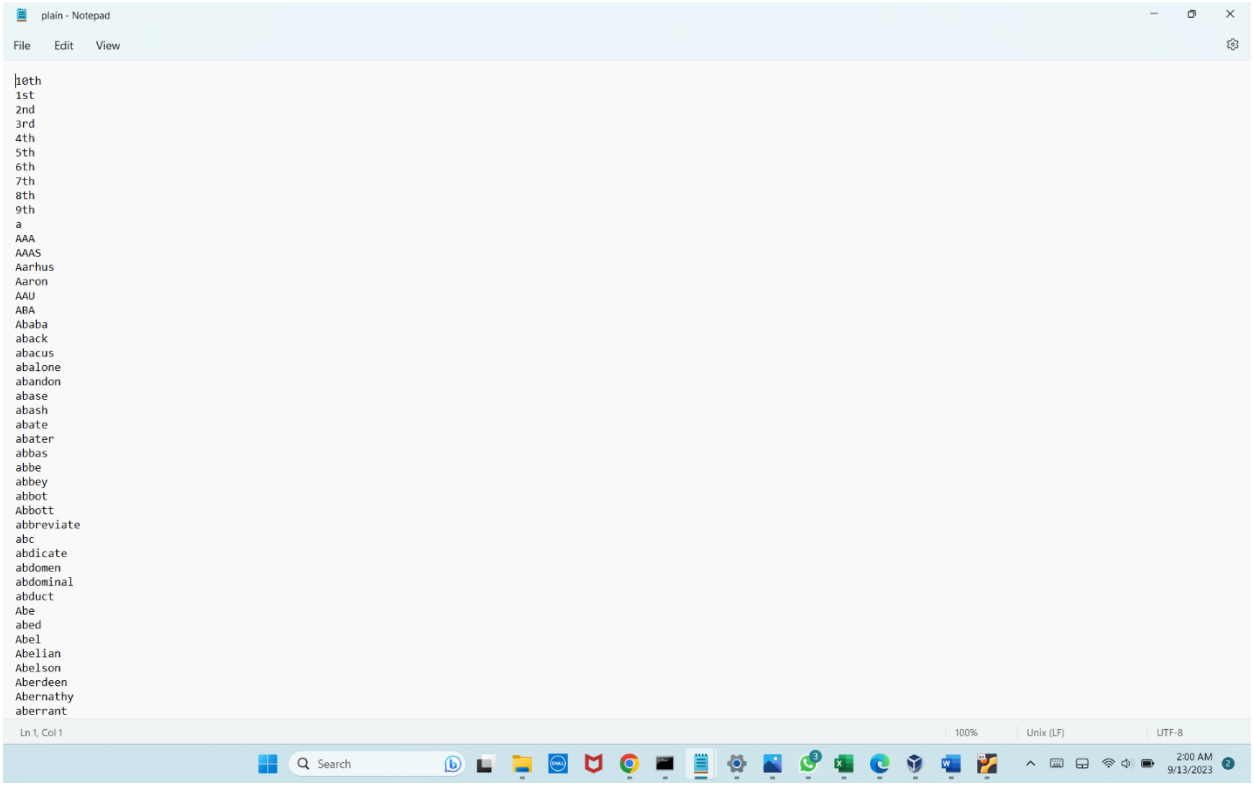
IN IT WAS A TOSSUP BETWEEN BOYHOOD AND THE EVENTUAL WINNER BIRDMAN
IN WITH LOTS OF EXPERTS BETTING ON THE REVENANT OR THE BIG SHORT THE
PRIZE WENT TO SPOTLIGHT LAST YEAR NEARLY ALL THE FORECASTERS DECLARED LA
LA LAND THE PRESUMPTIVE WINNER AND FOR TWO AND A HALF MINUTES THEY WERE CORRECT BEFORE AN ENVELOPE SNAFU WAS REVEALED AND THE RIGHTFUL WINNER MOONLIGHT WAS CROWNED

THIS YEAR AWARDS WATCHERS ARE UNEQUALLY DIVIDED BETWEEN THREE BILLBOARDS
OUTSIDE EBBING MISSOURI THE FAVORITE AND THE SHAPE OF WATER WHICH IS
THE BAGGERS PREDICTION WITH A FEW FORECASTING A HAIL MARY WIN FOR GET OUT

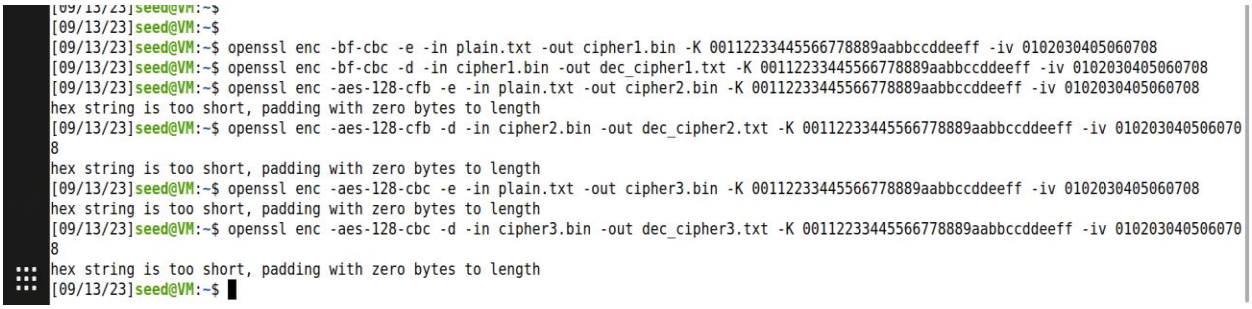
BUT ALL OF THOSE FILMS HAVE HISTORICAL OSCARVOTING PATTERNS AGAINST THEM THE
SHAPE OF WATER HAS NOMINATIONS MORE THAN ANY OTHER FILM AND WAS ALSO
NAMED THE YEARS BEST BY THE PRODUCERS AND DIRECTORS GUILDS YET IT WAS NOT
NOMINATED FOR A SCREEN ACTORS GUILD AWARD FOR BEST ENSEMBLE AND NO FILM HAS
WON BEST PICTURE WITHOUT PREVIOUSLY LANDING AT LEAST THE ACTORS NOMINATION
SINCE BRAVEHEART IN THIS YEAR THE BEST ENSEMBLE SAG ENDED UP GOING TO
THREE BILLBOARDS WHICH IS SIGNIFICANT BECAUSE ACTORS MAKE UP THE ACADEMYS
LARGEST BRANCH THAT FILM WHILE DIVISIVE ALSO WON THE BEST DRAMA GOLDEN GLOBE
AND THE BAFTA BUT ITS FILMMAKER MARTIN MCDONAGH WAS NOT NOMINATED FOR BEST
DIRECTOR AND APART FROM ARGO MOVIES THAT LAND BEST PICTURE WITHOUT ALSO EARNING BEST DIRECTOR NOMINATIONS ARE FEW AND FAR BETWEEN

Encryption using Different Ciphers and Modes

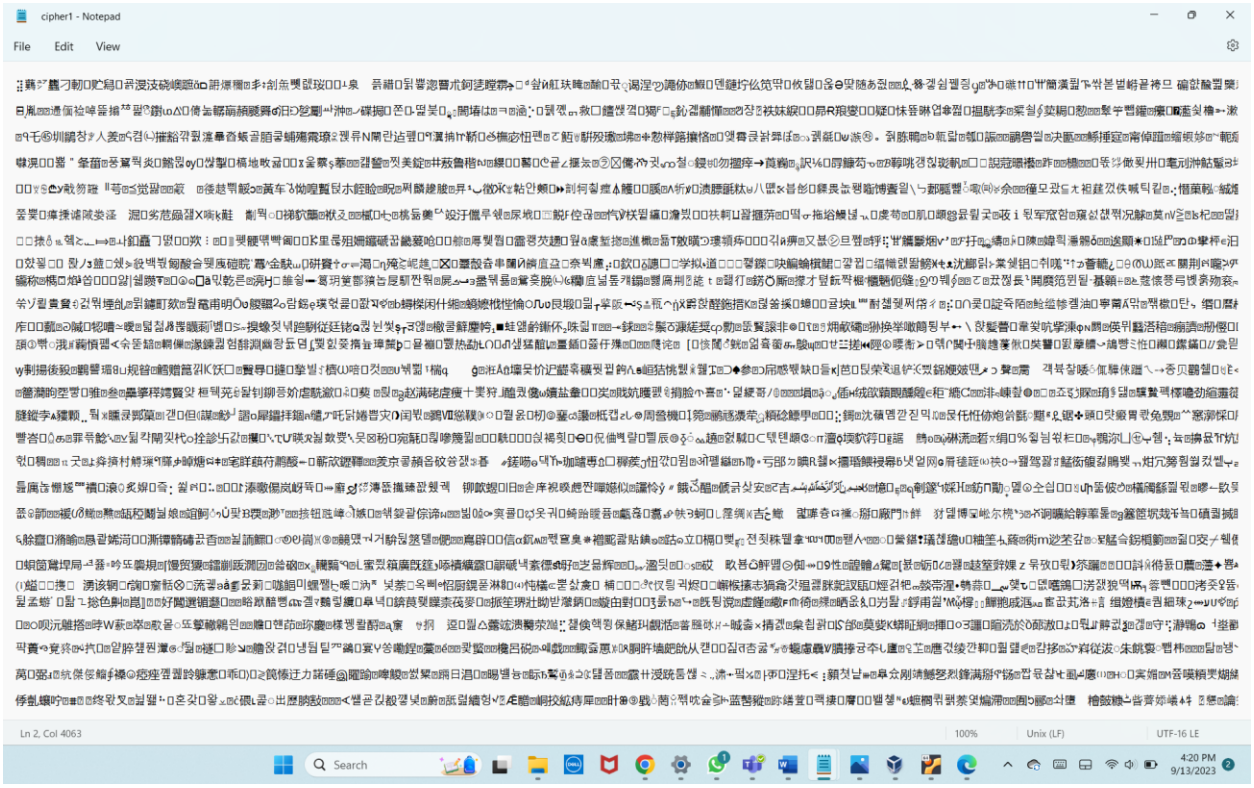
PLAIN TEXT :



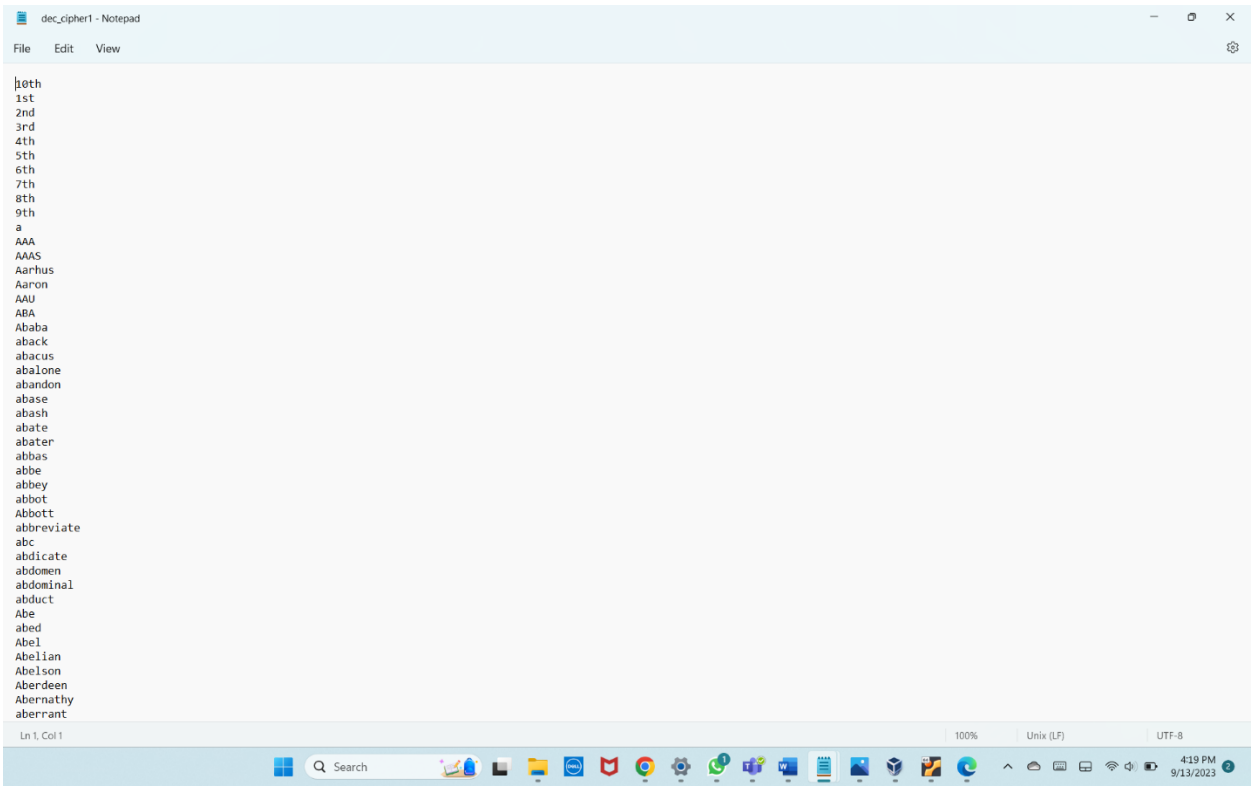
ENCRYPTION AND DECRYPTION CODE



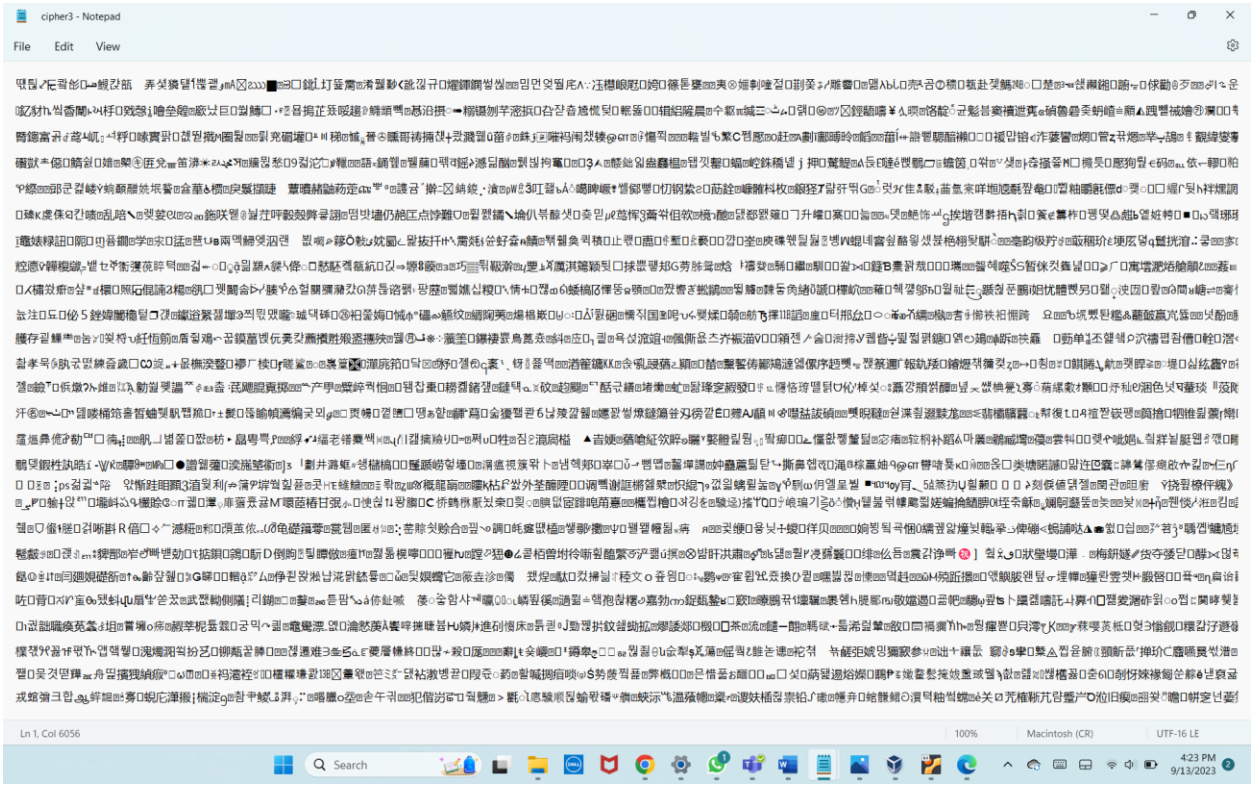
BLOW FISH IN CBC MODE ENCRYPTION – CIPHER TEXT



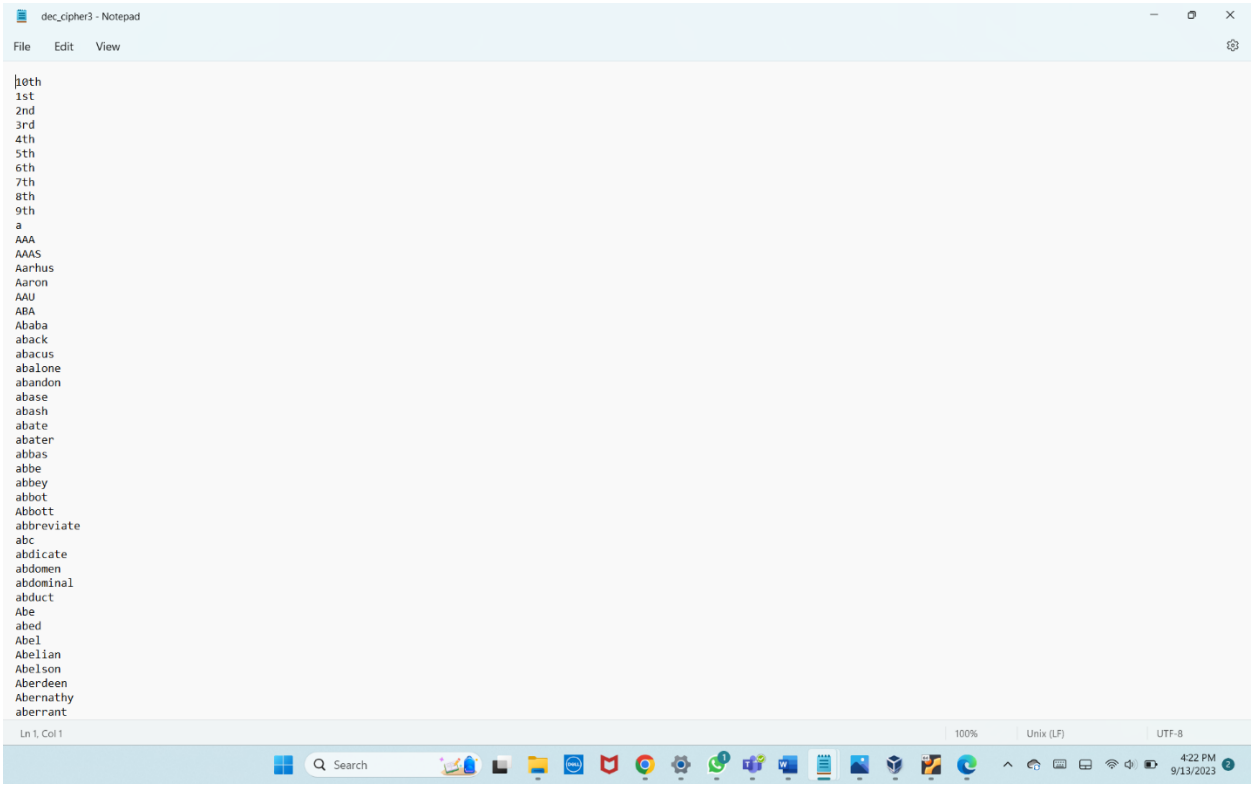
BLOW FISH IN CBC MODE DECRYPTION



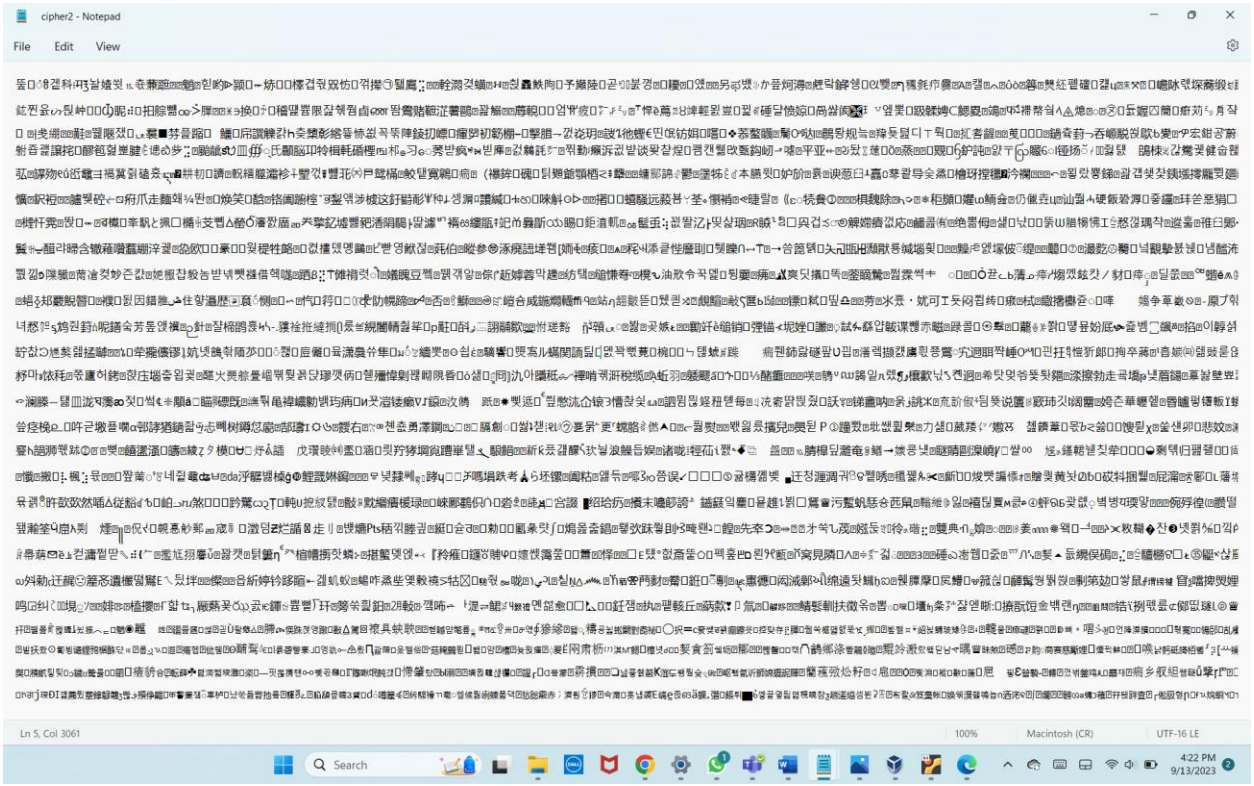
AES 128 CBC MODE ENCRYPTION – CIPHER TEXT



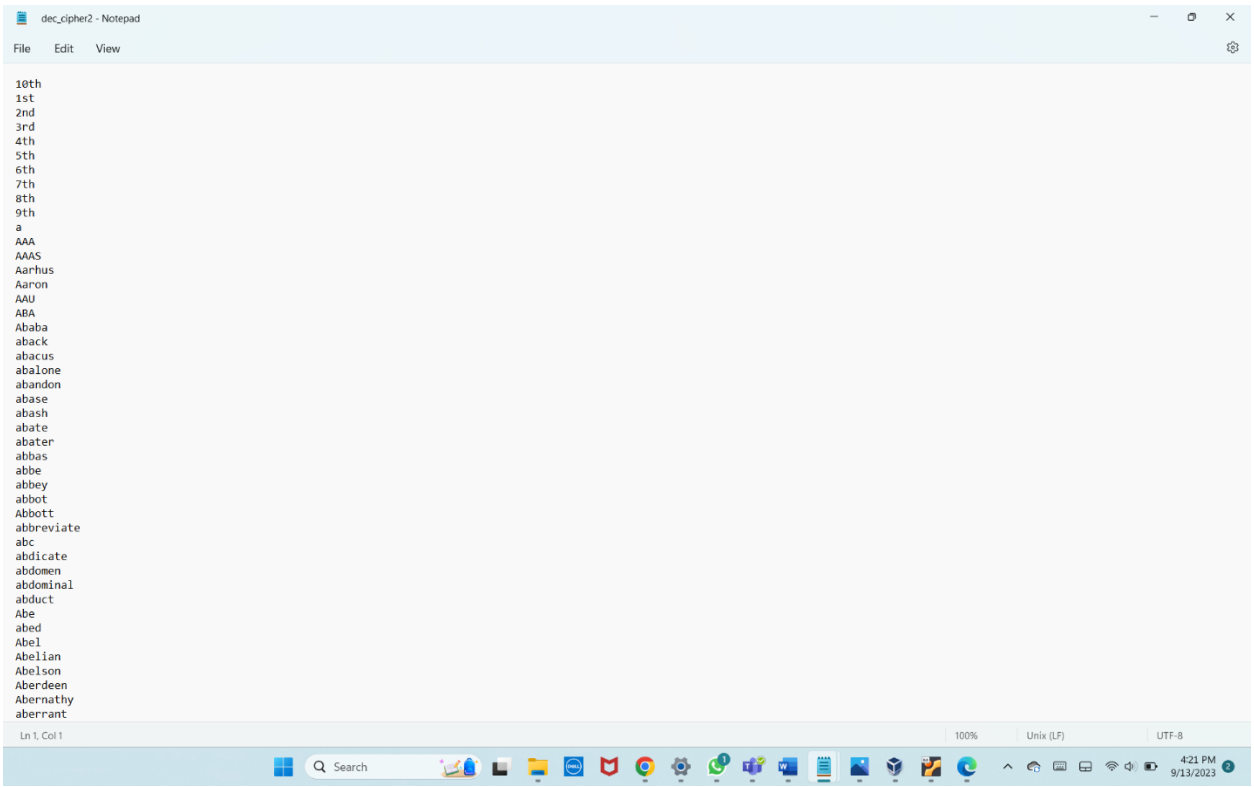
AES 128 CBC MODE DECRYPTION



AES – 128 CFB MODE ENCRYPTION – CIPHER TEXT



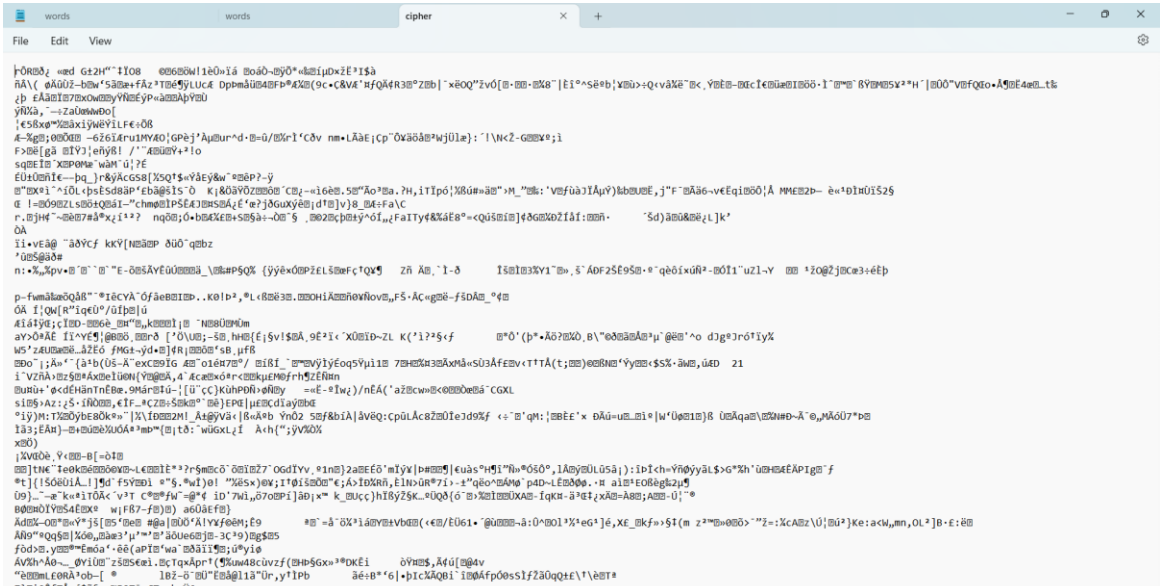
AES – 128 CFB MODE DECRYPTION



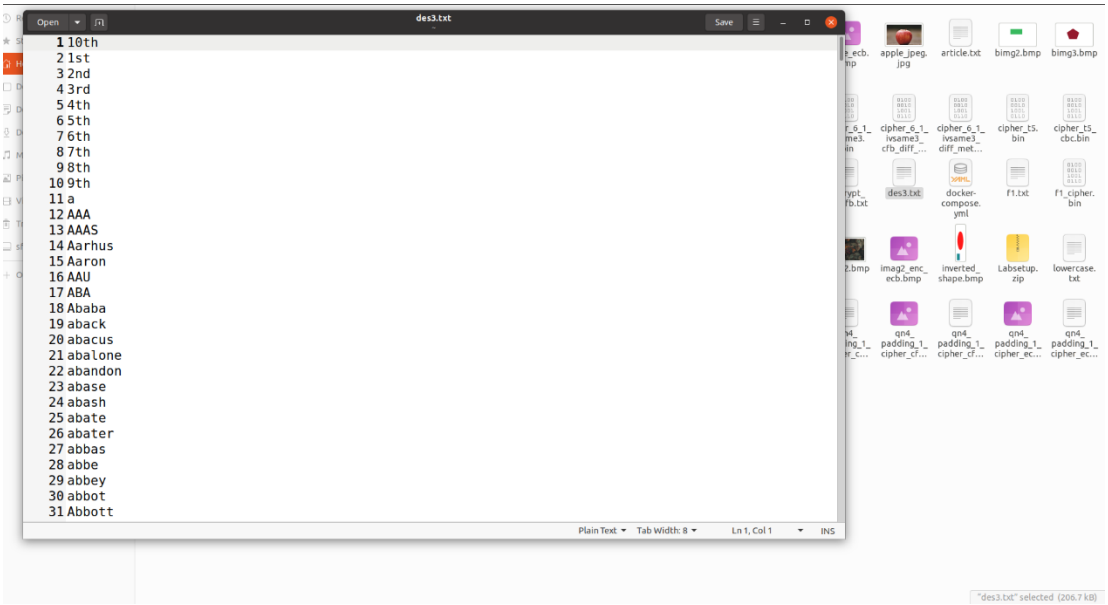
DES CODE SNIPPET:

```
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$ openssl enc -des3 -e -in plaintext.txt -out cipher.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$ openssl enc -des3 -d -in cipher.bin -out des3.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
[09/18/23]seed@VM:~$
```

DES ENCRYPTED TEXT :



DES DECRYPTED TEXT :



TASK DONE AND OBSERVATION :

Tried encryption and decryption for a plain text using

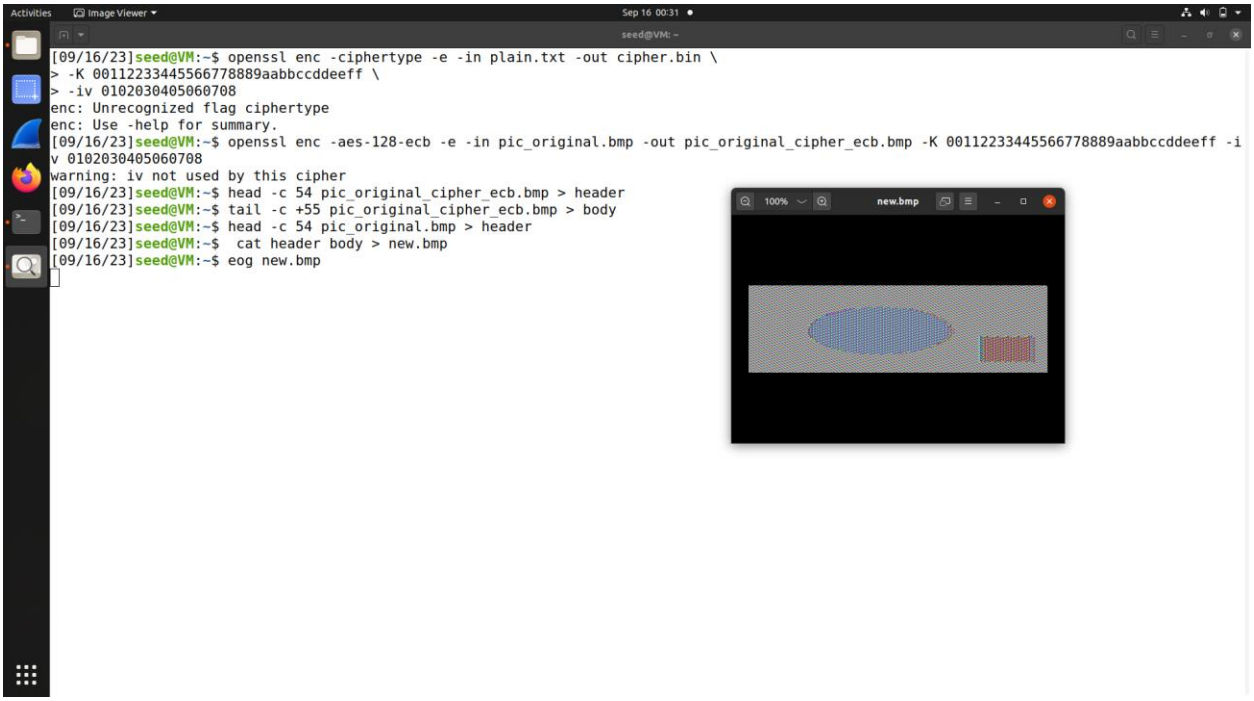
- AES-128 in CBC mode
- Blow Fish in CBC mode
- AES-128 IN CFB mode
- DES3

I was able to encrypt and decrypt by using the openssl command for the above ciphers and modes and understood the modes and flagoptions for encryption and decryption

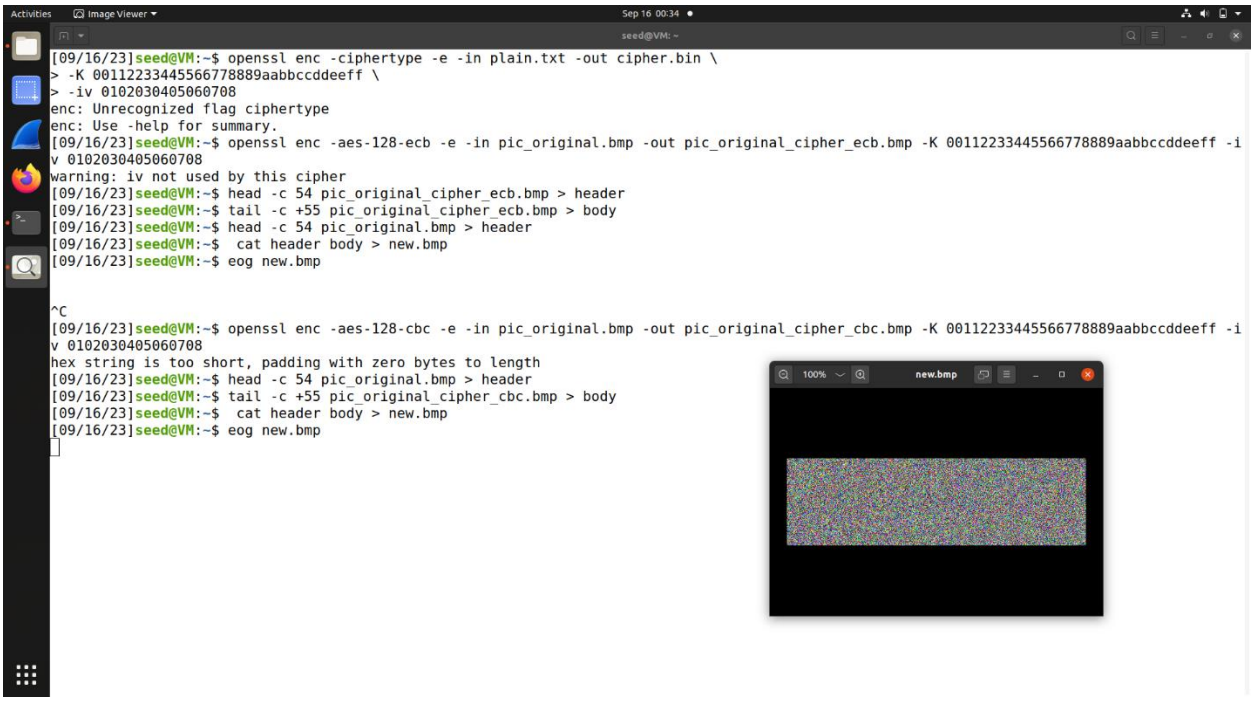
Encryption Mode – ECB vs. CBC

1. Image

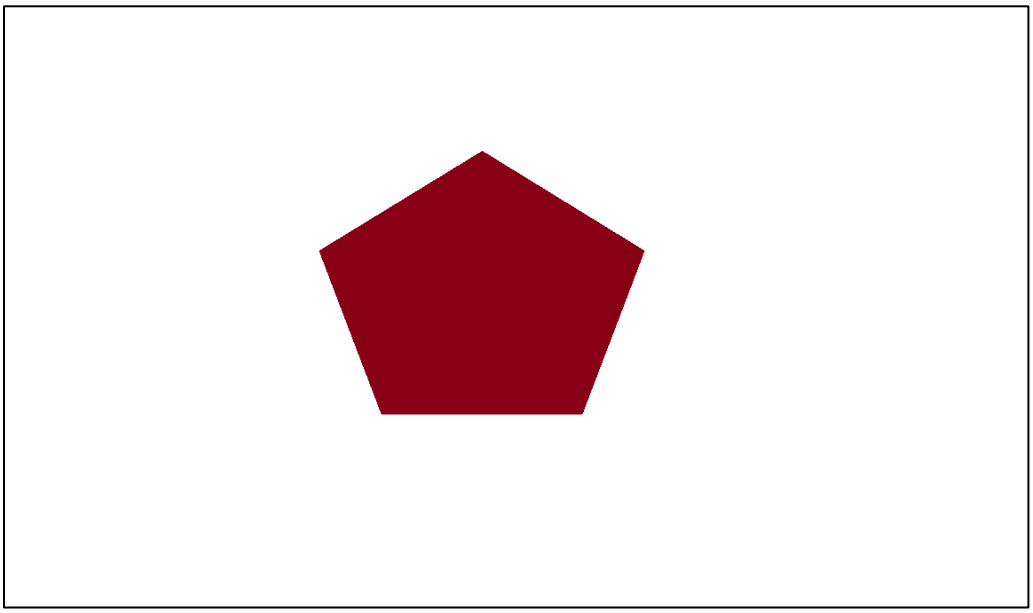
ENCRYPTED IMAGE IN ECB :



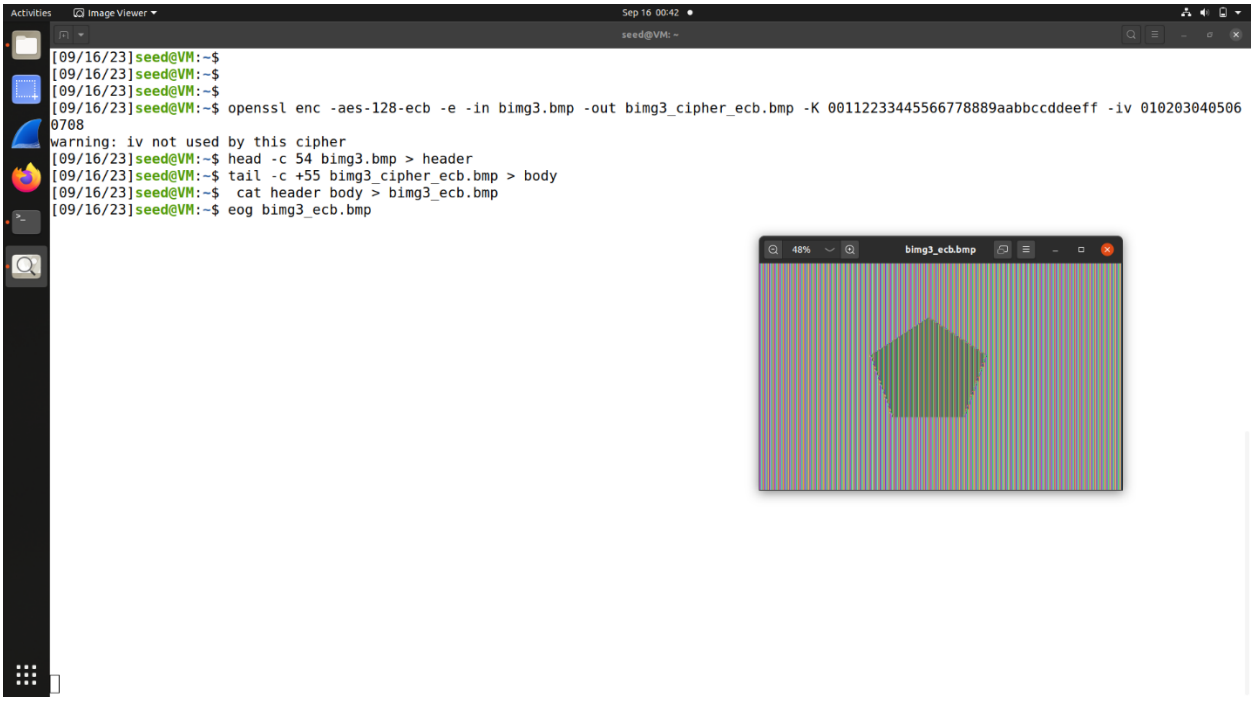
ENCRYPTED IMAGE IN CBC :



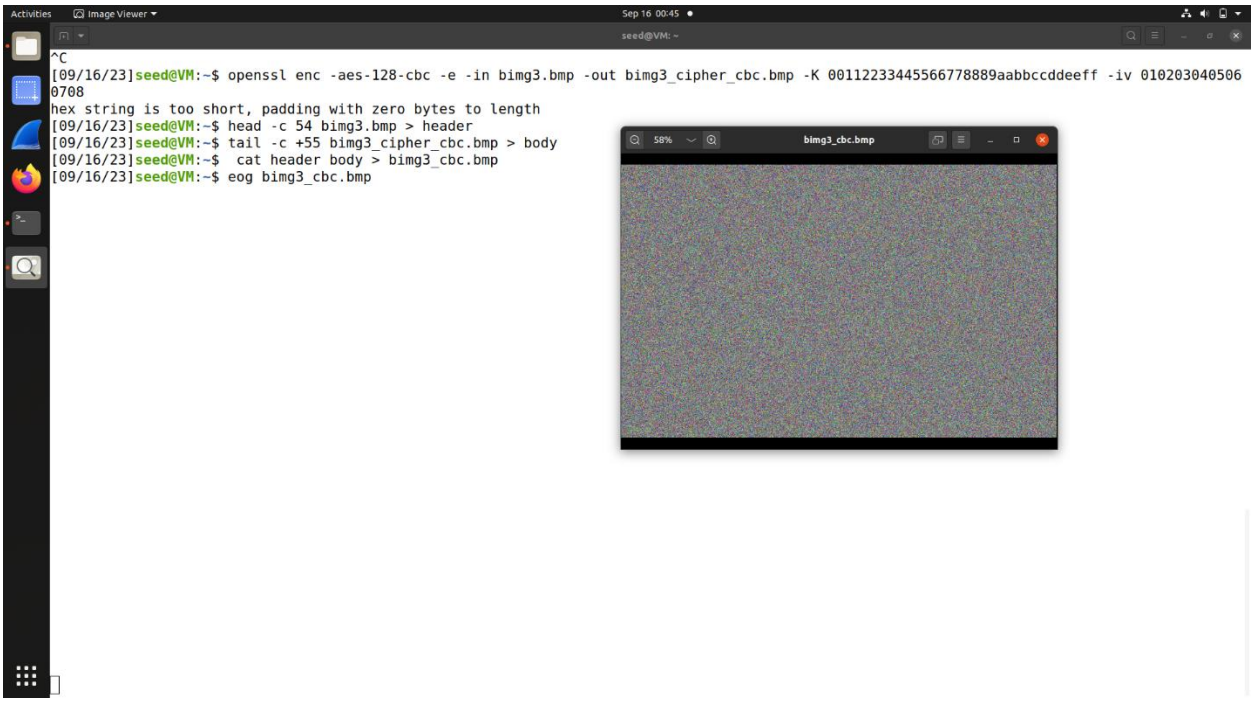
2. IMAGE



ENCRYPTED IMAGE IN ECB :



ENCRYPTED IMAGE IN CBC :

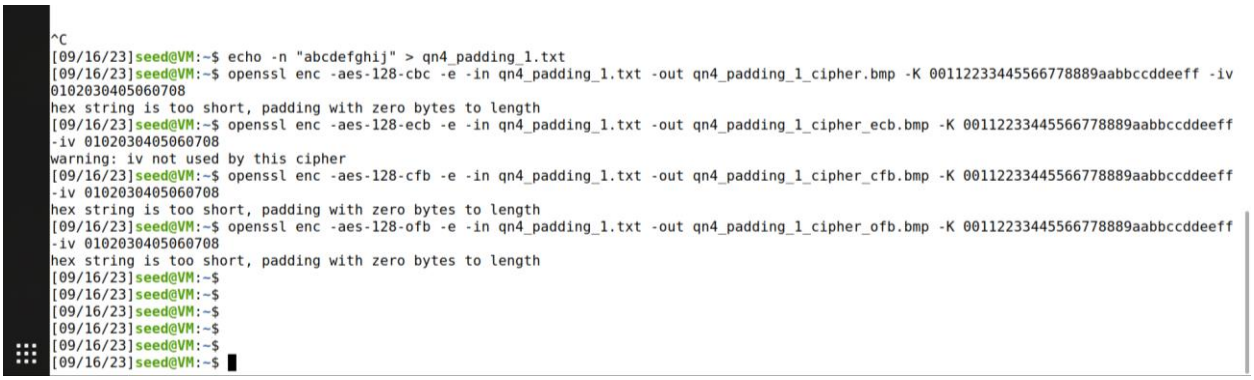


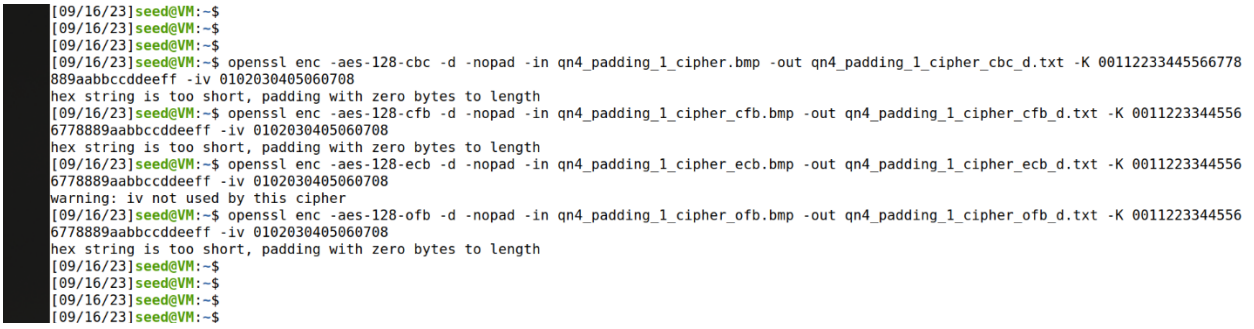
OBSERVATION :

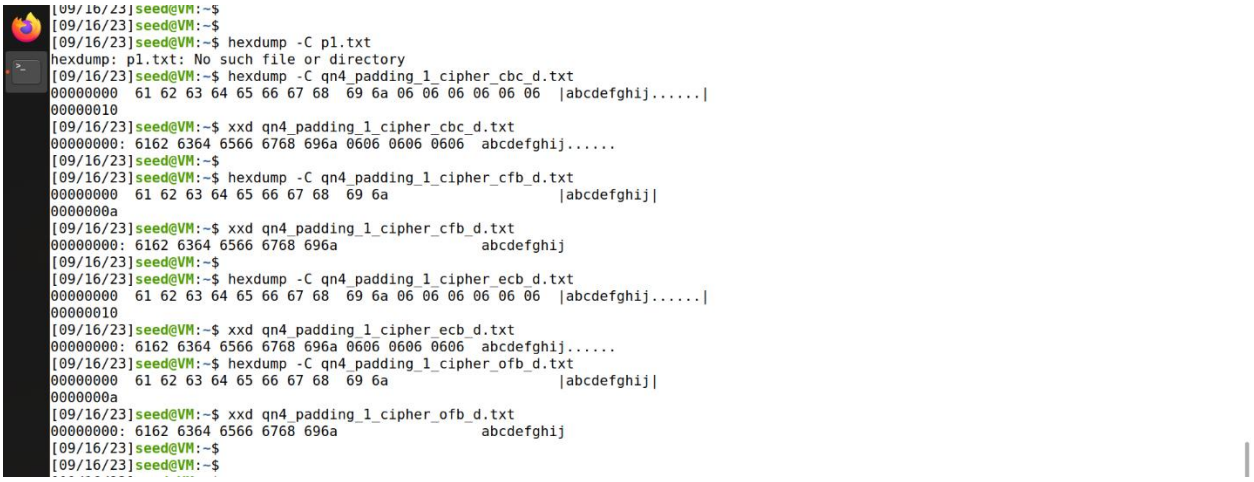
- It is observed the at the CBC mode of encryption is the best as compared to ECB mode of encryption.
- Because, In ECB mode we are able to predict the picture even after the encryption
- But we cant predict the picture in CBC mode
- After doing the encryption of given image, I tried with the new image and I could see the same behavior of CBC mode performing better than ECB mode

Padding

1.





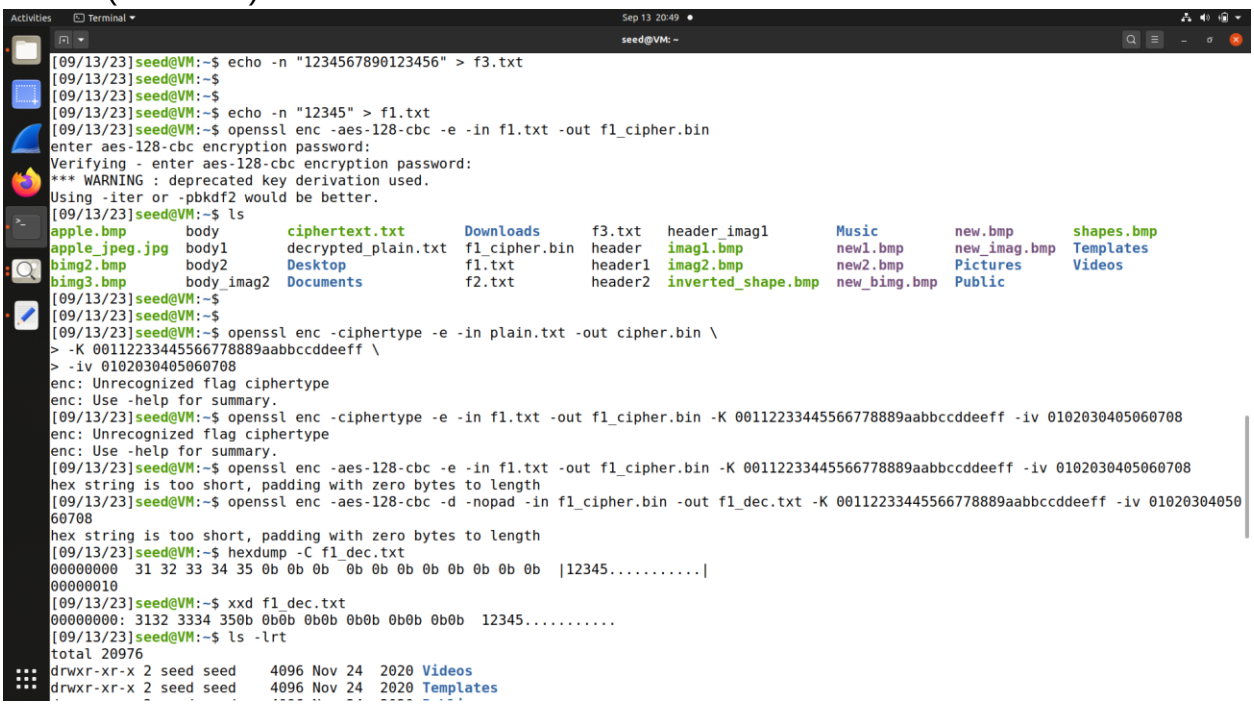


OBSERVATION:

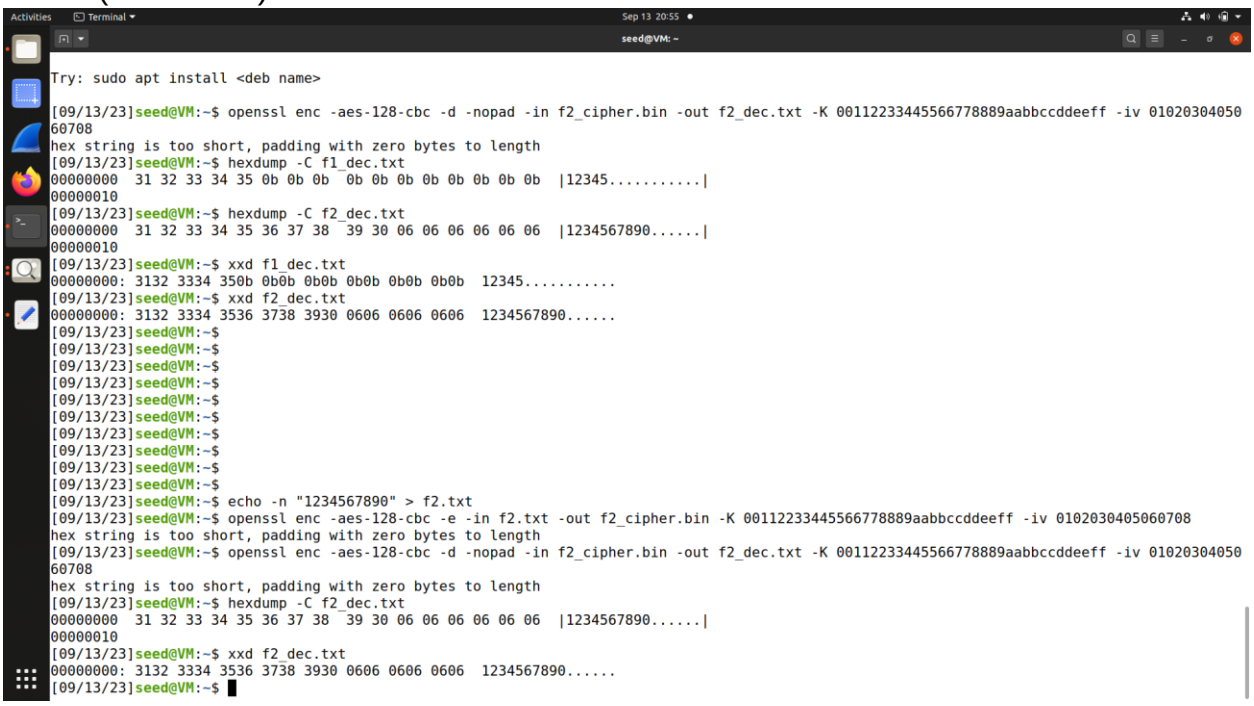
- ECB and CBC mode requires padding, because it requires fixed block size to encrypt the text. (It may be like 128 or 265).
- Here the padding bits 06 is appended the end.
- CFB and OFB does not require padding, because in these modes cipher text is always the same length as of the plain text as they do the operation byte by byte.

2.

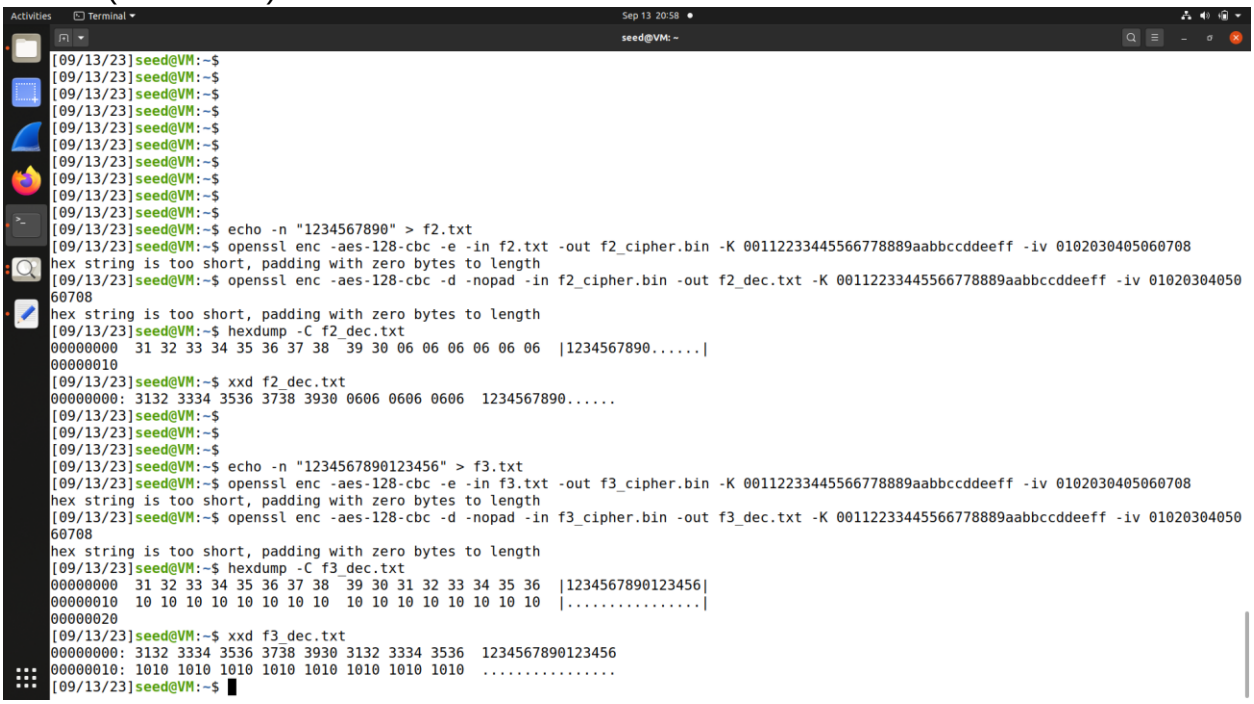
FILE 1 (5 BYTES):



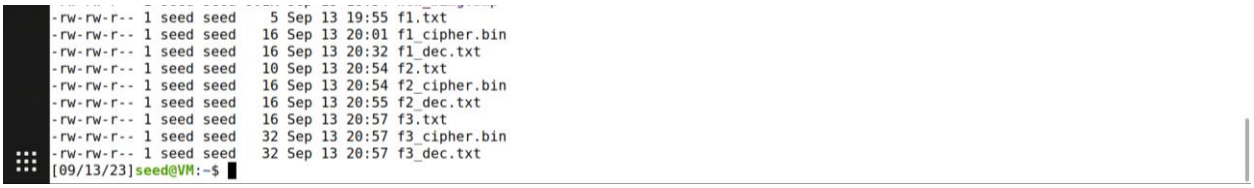
FILE2 (10 BYTES) :



FILE 3 (16 BYTES):



PLAIN TEXT , ENCRYPTED , DEENCRYPTED FILE SIZES :



Sizes of encrypted files are (in bytes)

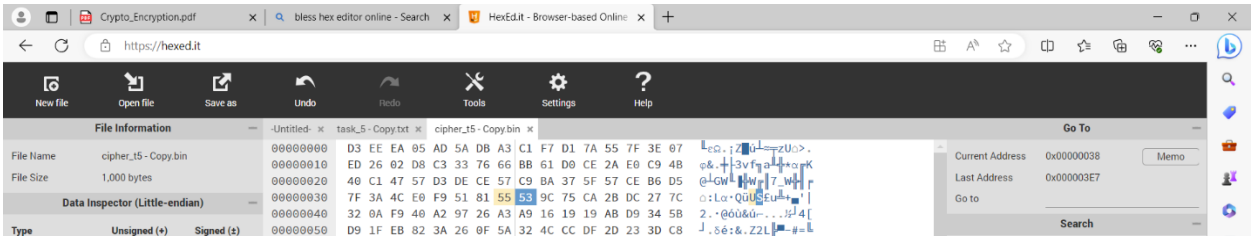
- 16 bytes for the file that contains 5 (bytes) characters
- 16 bytes for the file that contains 10 (bytes) characters
- 32 bytes for the file that contains 16 (bytes) characters

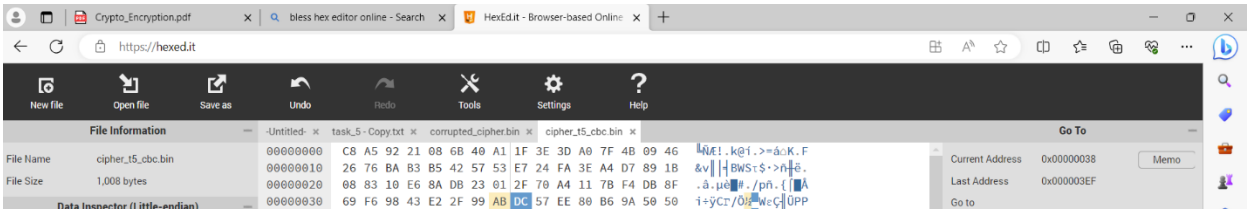
Padding bits

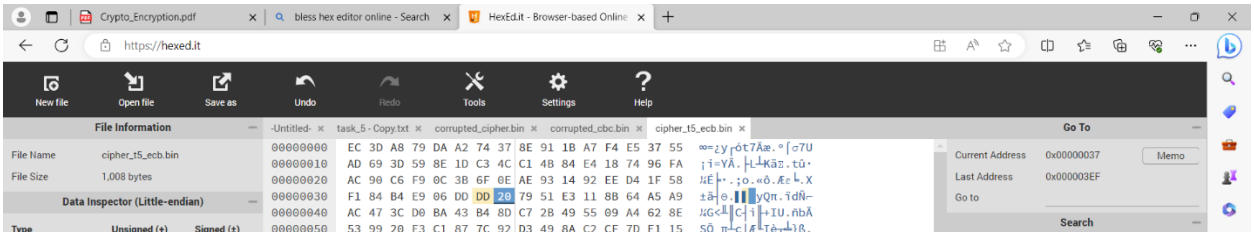
- In File 1 that contains 5 (bytes) characters, the padding bits 0b is appended to make it 128 bits
- In File 2 that contains 10 (bytes) characters, the padding bits 06 is appended to make it 128 bits
- In File 3 that contains 16 (bytes) characters, no padding bits are added as it already 128 bits long

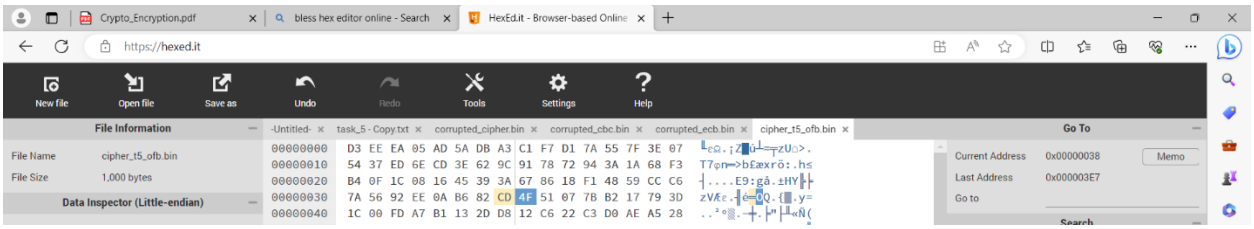
Error Propagation – Corrupted Cipher Text

CORRUPTING THE TEXT AT 55TH BYTE

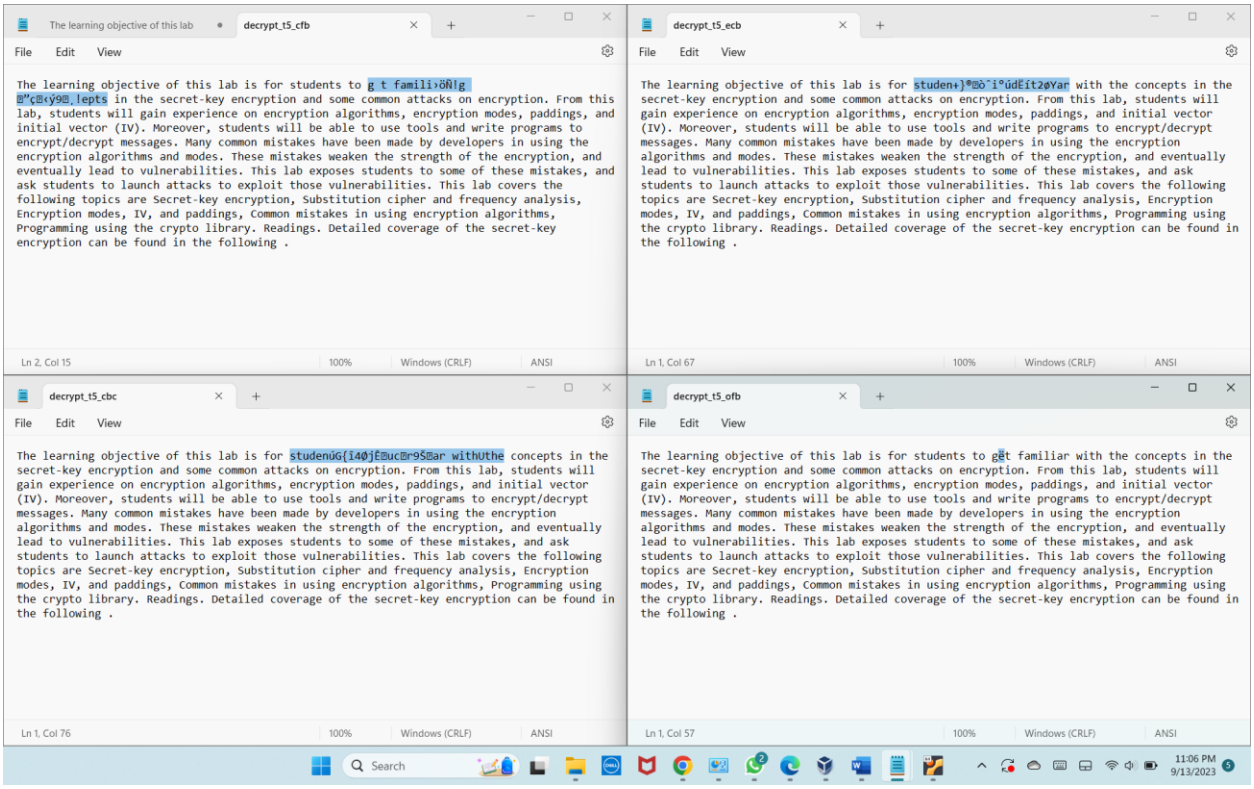




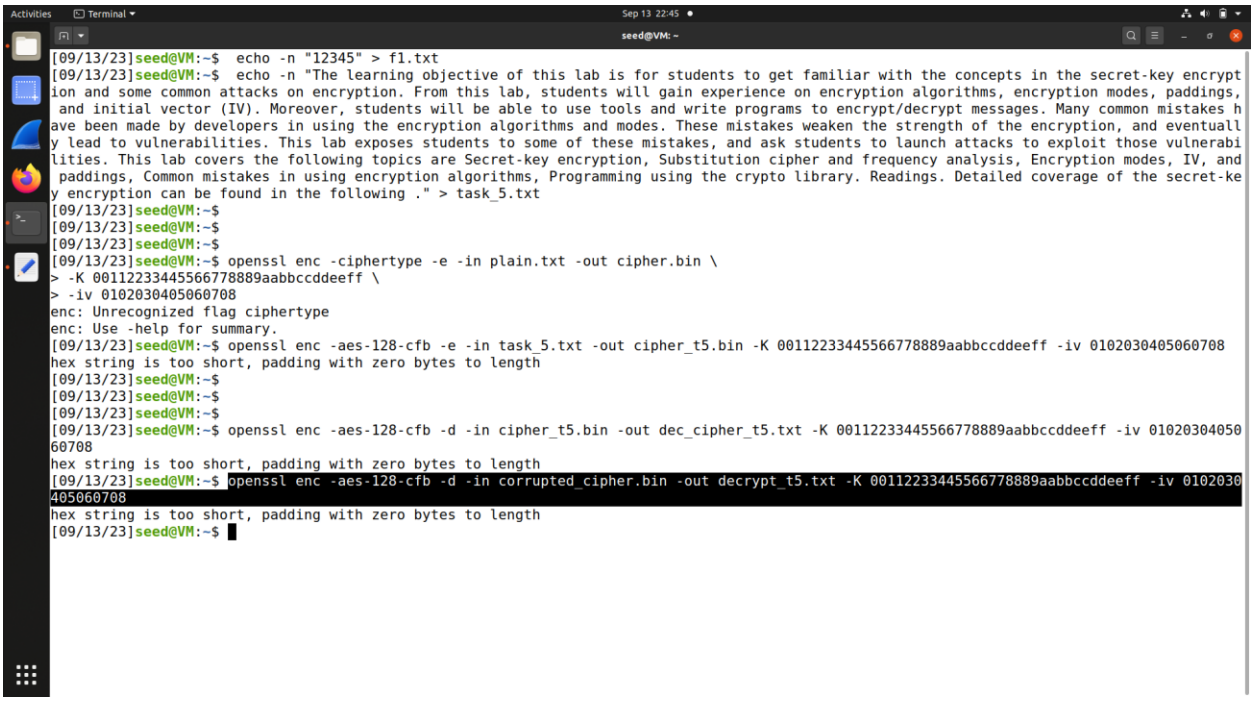


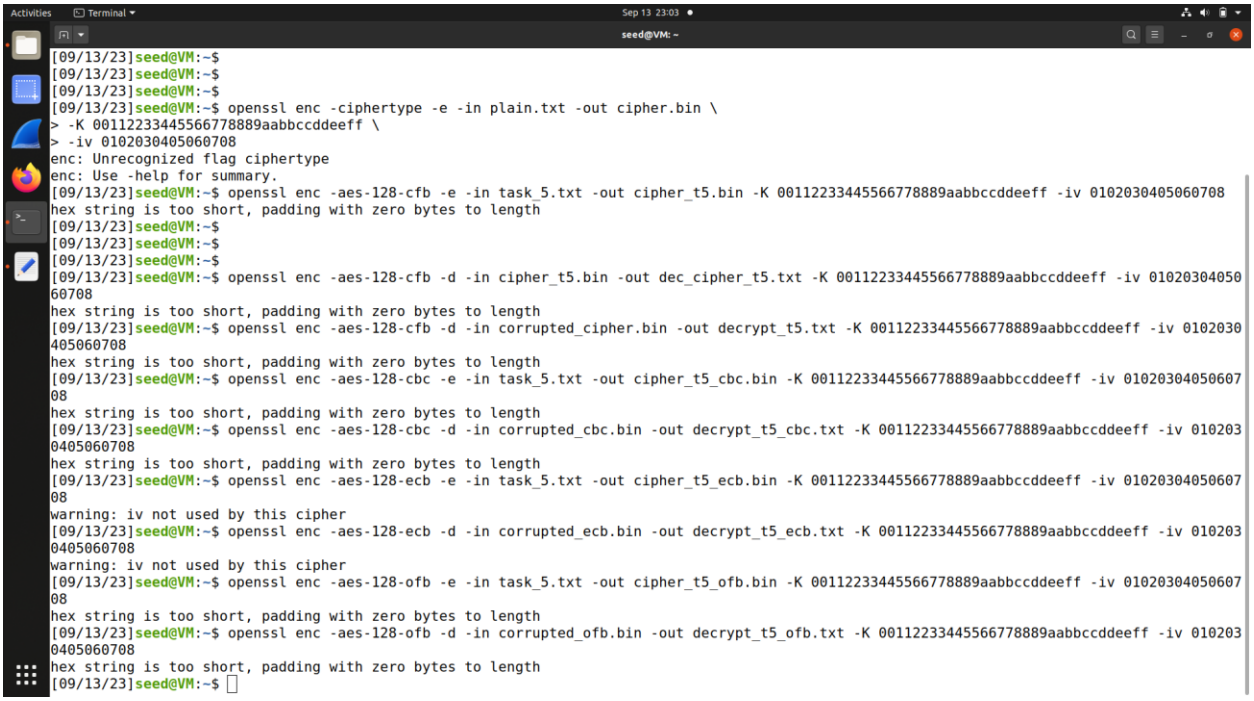


CORRUPTED TEXT FOR CFB, CBC,ECB, OFB RESPECTIVELY



LINUX SCRIPTS :



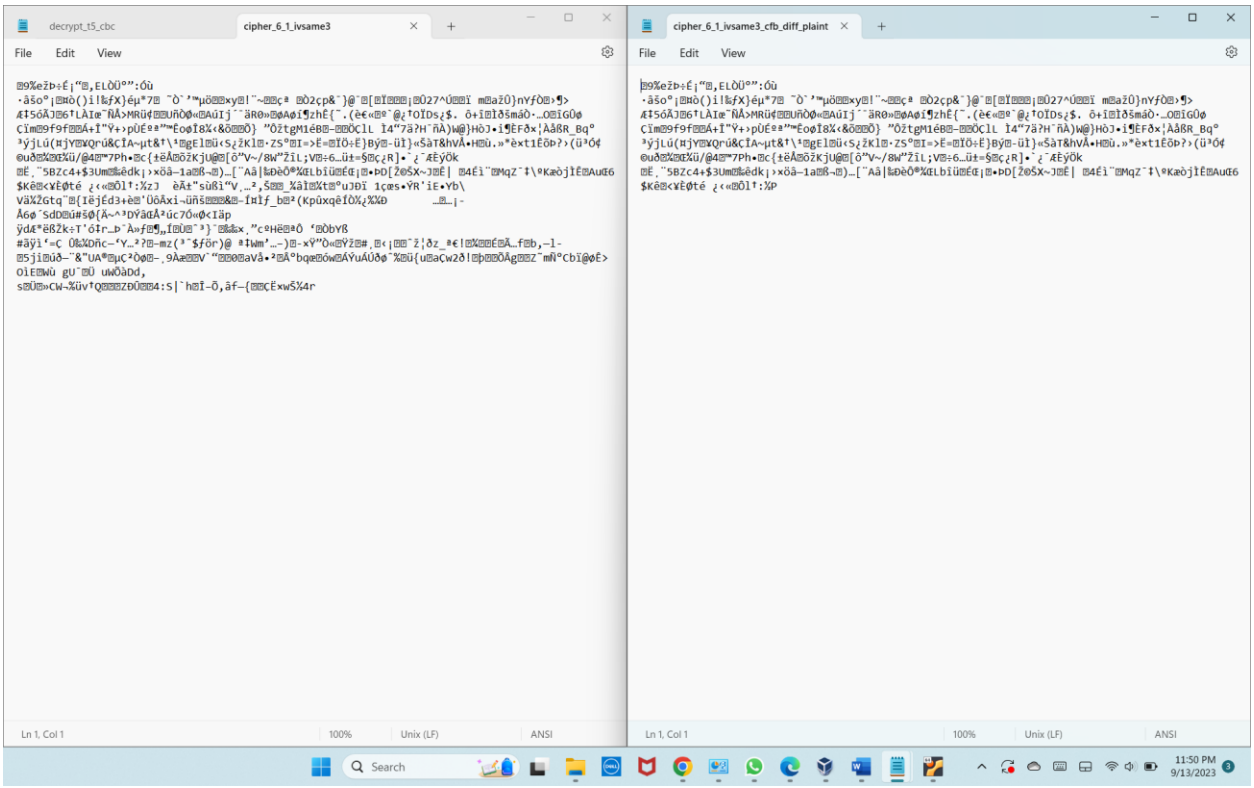


TASK DONE AND OBSERVATION

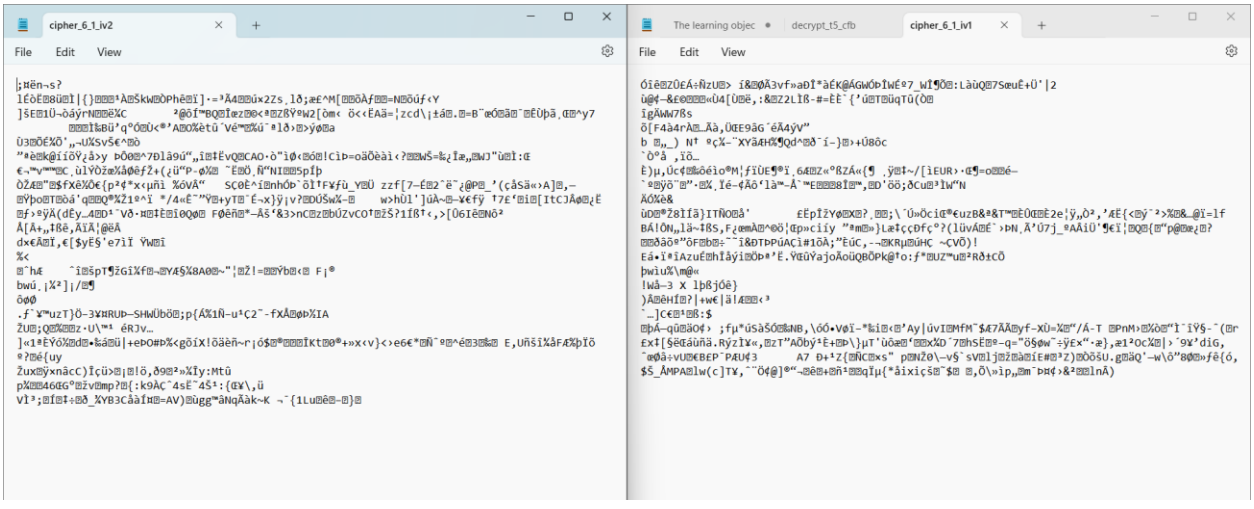
- Created a file of 1000 bytes long and encrypted it using AES 128 Cipher
- Corrupted the 55th bit using hexEdit online tool
- Then decrypted the corrupted file
- It is observed that some characters were in non readable form and in case of OFB mode only the corresponding byte was unreadable and others were decrypted properly
- Hence Im able to infer that the OFB mode is the least affected mode to corruption out of the above modes

Initial Vector (IV) and Common Mistakes

Case1 : SAME IV



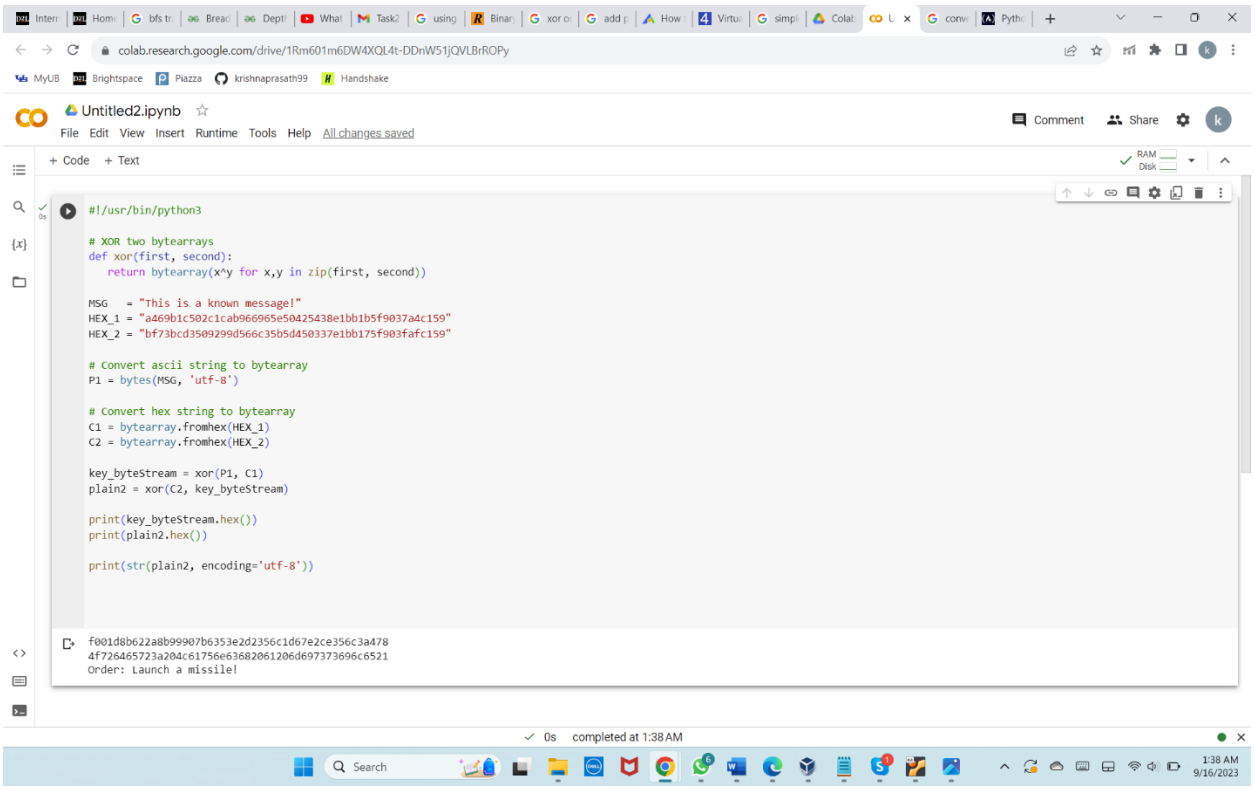
Case 2 :DIFFERENT IV



OBSERVATION :

- In case1 while using the same IV, some of the encrypted text's pattern are repeated.
- So it is easier for the attacker to predict the pattern of the cipher and find the text easily.
- But when we used different IV (In case 2), even for the same text, the cipher text differs.
- Hence, we can understand that the IVs should be different

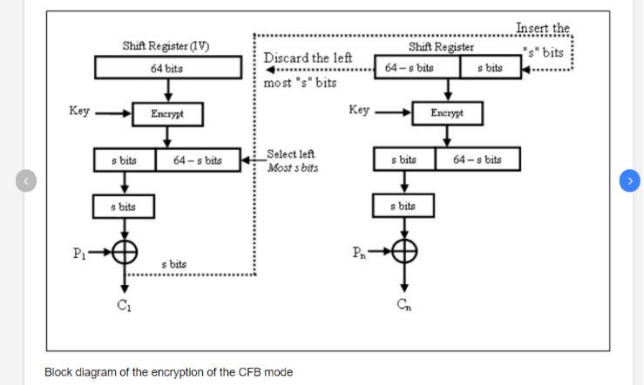
Common Mistake: Use of Same IV



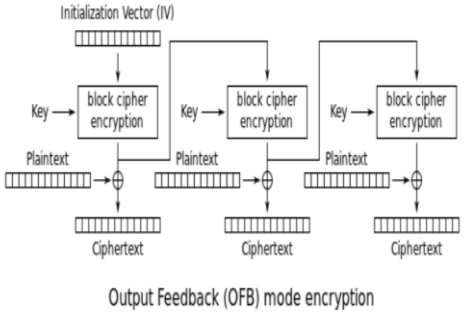
Plain text P2 is `Order: Launch a missile!`

TASK DONE AND OBSERVATION :

- As the xor operation is commutative, if we do the xor operation on plaintext (in byte array) with its corresponding cipher text we could get the key, because cipher text is obtained by doing the xor operation on plain text and key.
- Hence we can find the key.
- After finding the key, we can do the xor operation with key and other cipher text to find the plain text



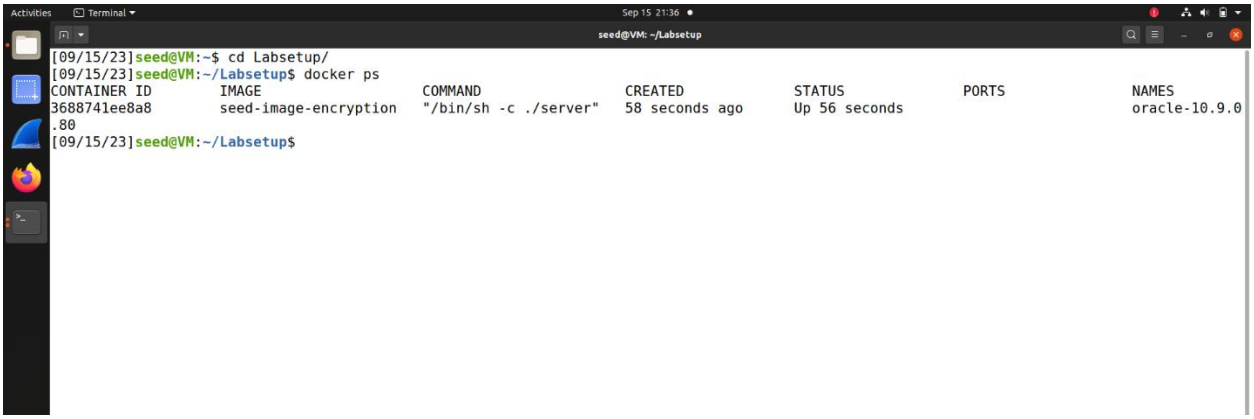
Block diagram of the encryption of the CFB mode

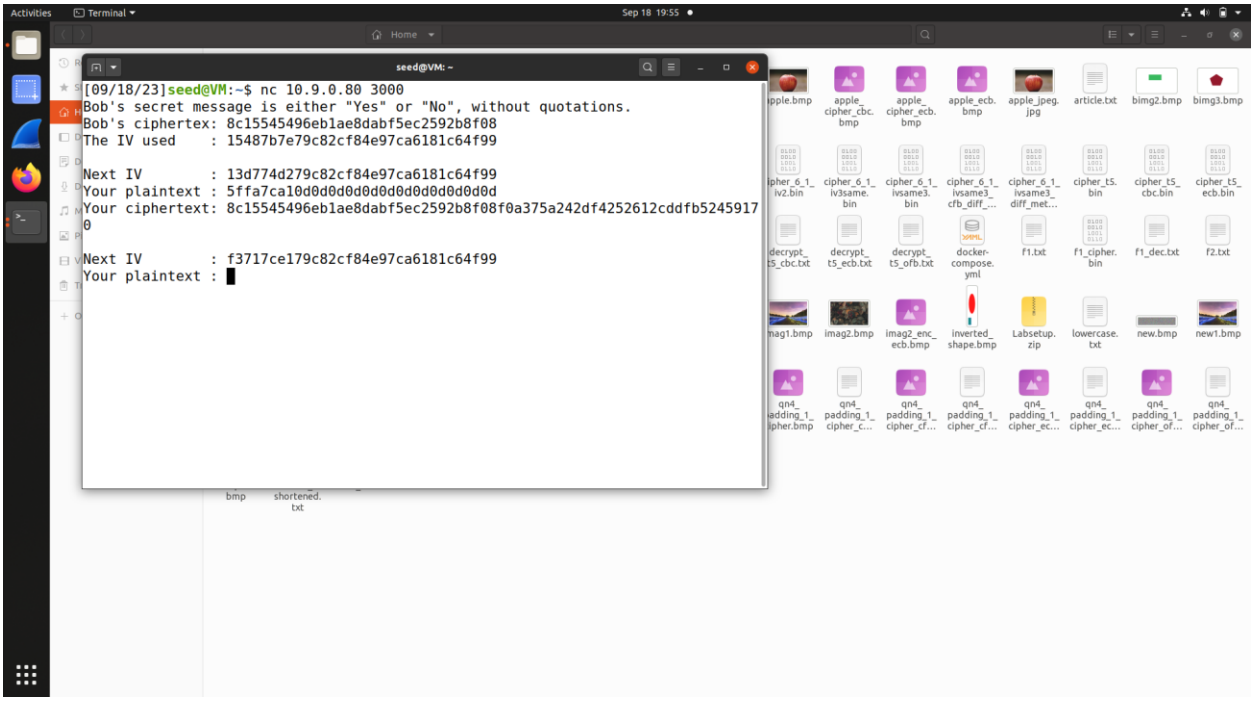


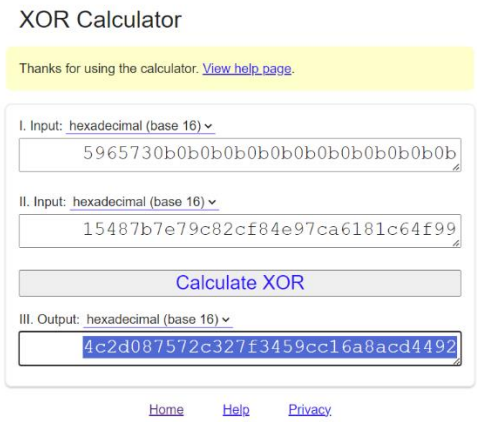
Output Feedback (OFB) mode encryption

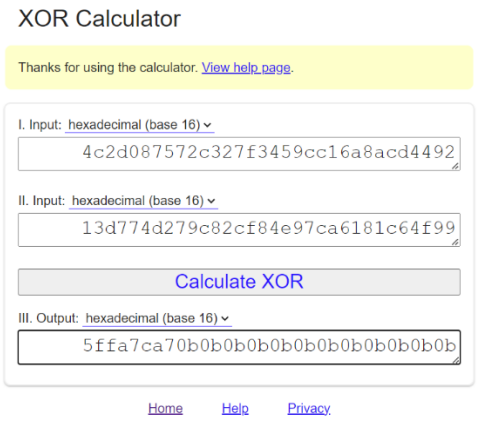
- From the block diagrams of CFB and OFB mode we can understand that the, for the first 64 bits of data the IV is same,
- Then the encrypted key is given as IV for next 64 bits in OFB
- In CFB its xor with the Plain text and its given as input of IV for the next 64 bits.
- Hence if we replace OFB with CFB in this experiment, **the first 64 bits of data can be revealed** as there is no change of logic in the IV

Common Mistake: Use a Predictable IV









TASK DONE AND OBSERVATION :

- As the cipher is AES-128-CBC mode, and the plain text is either ‘Yes’ or ‘No’, I tried to convert the string ‘Yes’ to hexadecimal form and added the padding bits at the end.
- Then I did the xor operation on current IV with the padded data as xor operation is commutative
- Then did the xor operation of the result with the next IV, so that if it is passed as the plain text it give the same cipher text.
- Thus form this it can be concluded that the plain text is ‘Yes’
- I was able to understand that if the attacker have the current IV, next IV and the

Programming using the Crypto Library AES-128-CBC

Source code :

```
from Crypto.Cipher import *
import binascii

plaintext = "This is a top secret."
given_ciphertext="764aa26b55a4da654df6b19e4bce00f4ed05e09346fb0e762583cb7da
2ac93a2"
init_vector="aabbccddeeff00998877665544332211"

pad_size = 32-len(plaintext)
padding = chr(pad_size) * (pad_size)
plaintext = plaintext + padding
print("Plain text   ---------> ",plaintext)


iv_byte_arr=bytearray.fromhex(init_vector)
print("Hex iv -------->",iv_byte_arr);
f=open("words.txt","r")
allKeys=f.read().splitlines()

for key in allKeys:

    if len(key)==0:
        continue

    if len(key)>16:
        continue


    elif len(key)%16 !=0:
        key_length=len (key)
        while key_length %16!=0:
            key =key+"#"
            key_length = key_length+1

    new_ciphertext = AES.new(bytes (key, "utf-8"), AES. MODE_CBC, iv_byte_arr)
    byte_cipher = new_ciphertext.encrypt (bytes (plaintext, "utf-8"))
    hex_cipher_text=str(binascii.hexlify(byte_cipher).decode())

    if (hex_cipher_text==given_ciphertext):
```

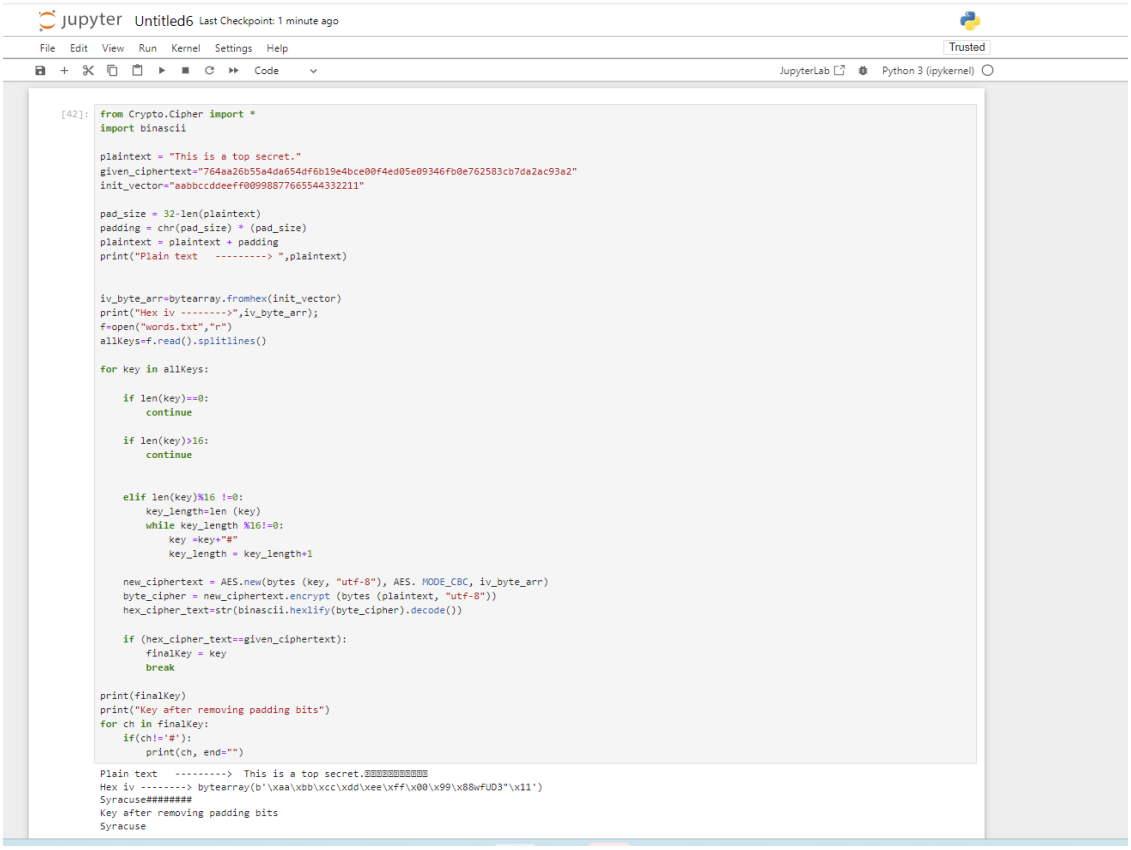
```
        finalKey = key
        break

print(finalKey)
print("Key after removing padding bits")
for ch in finalKey:
    if(ch!='#'):
        print(ch, end="")
```

Output :

```
Plain text   ---------> This is a top secret.
Hex iv --------> bytearray(b'\xaa\xbb\xcc\xdd\xee\xff\x00\x99\x88wfUD3"\x11')
Syracuse########
Key after removing padding bits
Syracuse
```

Hence the key is **Syracuse**



Code Logic :

- Importing the necessary cipher library
- Add 11 byte padding (32 - 21) to the plain text, 21 is length of plain text
- Convert the given hexadecimal initial vector to byte array
- Read all the keys from the given file
- For each key, validate the size, add ‘#’ to make it 16 bytes length (if length < 16)
- Encrypt the plain text using each key
- If the encrypted cipher text matches with the given cipher text, break the loop and exit and print the key

Output :

Hence we are able to get the Key as Syracuse with the additional padded bits ‘#’ at the end. As the length of key is 8 (8 padding bits are added). Then padding bits are removed and the key is displayed.