



Secret-Key Encryption Lab Report

Emre Can Tüzer

Date: November 4, 2024

BGK-503

Kemal Bıçakçı

Task 1: Frequency Analysis

Utilized a Python script to analyze the frequency of n-grams in a given ciphertext.

```
[10/26/24] seed@VM:~/.../Files$ freq.py
-----
1-gram (top 20):
n: 488
y: 373
v: 348
x: 291
u: 280
q: 276
m: 264
h: 235
t: 183
i: 166
p: 156
a: 116
c: 104
z: 95
l: 90
g: 83
b: 83
r: 82
e: 76
d: 59
-----
2-gram (top 20):
yt: 115
tn: 89
mu: 74
nh: 58
vh: 57
hn: 57
vu: 56
nq: 53
xu: 52
up: 46
```

Starting with common letters, the letter "ytn" is a good choice for the English word "THE" because it appears frequently.

ytn - THE

ytnu - THEu

ytnd - THEd

ytnhn - THEhE (*h would be R*)

hnden yürüdüm

Inhn - IERE - (*I would be W*)

ytvy - THvT - (*v would be A*)

vhn - vRE - ARE

hnp - REp - RED

vup = ANp (*p would be d*)

ytnmh - THEmh - THEmR - (m would be i)

vuxytnh - vuxTHEh - AuxTHER - (ANOTHER) -- (u=N , x=O)

ivupmur - iANDINr - Maybe LANDING (i = L, r=g)

ivup - LAND

Since I was unable to locate any more, I will decipher these words and look over the text as a whole.

**tr 'ytnhlvmuxpirszkqb' 'THERWAINODLGKUXSF' < decryptedtext_quarter.txt >
decryptedtext_half.txt**

```
[10/26/24]seed@VM:~/.../Files$ tr 'ytnhlvmuxpir' 'THERWAINODLG' < ciphertext.txt > decryptedtext_quarter.txt
[10/26/24]seed@VM:~/.../Files$ tr 'ytnhlvmuxpirszkqb' 'THERWAINODLGKUXSF' < decryptedtext_quarter.txt > decryptedtext_half.txt
[10/26/24]seed@VM:~/.../Files$
```

ciphertext.txt decryptedtext_quarter.txt

```
1 THE OgAArq TzRN ON qzNDAd WHIAH qEcq Ag0ZT RIGHT ABTER THiq LONG qRANGE
2 AWARDq TRIE THE gAGGER BEELq LiSE THE oNAGENARIAN TOO
3
4 THE AWARDq RAAE WAQ qOOSENDED qd THE DEC1qE Ob HARFEd WEINSTEIN AT ITq 0zTqET
5 AND THE EeeARENT IcelQOION Ob HQ b1Lc abceANd AT THE END AND IT WAq qHAeED gd
6 THE ECERGENAe Ob cETOQ TlEq ze glAsqOWN eOLTTIq ARcaNDI AaTTI1qC AND
7 A NATIVE JOURNALqATION Ad gRIED AND caq Ag A BETER DREAM AGO WHETHER THERE
8 OXOG TO qB AEREEq THE qoAqRq WERE COED TO THE DIBRT WEESEND IN CARAH TO
9 EKTRA LONG gEAZqE THE OgAArq WERE COED TO THE DIBRT WEESEND IN CARAH TO
10 AFAOID a0NbLIATING WITH THE aL0QING aEREcOND Ob THE WINTER Oldcelaq THAnsq
11 edONGaHANG
12
13 ONE sIG jEsTION qPROFOUNDING THIn dEARq AaDeed AwhRde Ta HOW On Tb THE
14 aEREcOND WILL ADDREEq cETOQ EqEaTALLD ABTER THE GOLDEN GLOqEq WHIAH gEAce
15 A ozqILANT a0CING0Zt aRTq b0R TICEq ze THE c0FeCENT qeEARHEADED gd
16 e0WERDZ HOLLOWWOOD WOMEN WHO HELED RA1qE CILLIONq Ob DOLLARq TO NIGHT qEKeAL
17 HARAquCENT AR0zN THE w0zNTd
18
19 qIGNALING THEIR qzeORT GOLDEN GLOqEq ATTENDEEq qWAThED THEc0ELfEq IN qLAas
20 qoERTED LAEL elNq AND qzNDed 0bd Ag0ZT qEK1qT oMER IcaGLANqOb bRoC THE RED
21 SARGET AND THE STAGE ON THE AIR E WAQ qALLED Ob Ag1qT ead INq ZITId ABTER
22 THE STAGE AND THE PLAYERS CARRIED THEM OUT THE CARRETAH THE qoAqRq MAM qAR
23 LEq THAN A CALq qHONq AND DURING THE qRECOND NATALIS q0TCAN TOOs A qL1NT
24 AND qATIqoING DIG AT THE ALLqALE R0qTER Ob NOcINATED DIRECTOrq HOW a0zLD
25 THAT qE ToeEq
26
27 Aq IT TzRNq 0zT LEAQq IN TERcq Ob THE OgAArq IT eRoqgqLd WONT qE
28
29 WOCEN INFOLFED IN TICEq ze qAID THAT ALTH0ZGH THE GLOqEq qIGNIdIED THE
30 INITIATIFe LAzNah THED NEFER INTENDED IT TO qZqD AN AWARDq qAqD
31 qZqD qZqD OR qZqD qZqD
32 qZqD qZqD
33 qZqD qZqD
```

WEEsEND - s is K

AROzND - z is U

EkTRA - k is X

qTAGE - q is S

AbTER - b is F

**tr 'ytnhlvmuxpirszkqb' 'THERWAINODLGKUXSF' < decryptedtext_quarter.txt >
decryptedtext_half.txt**

aOUNTRd - aOUNTRd - a is C, d is Y

aOULD - Could - a is C

jUIT - Quit - j is Q

The last command to fully decode the text is:

tr 'vgapnbrtmosicuxejhqyzflkdw' 'ABCDEFGHIJKLMNPQRSTUVWXYZ' < ciphertext.txt > plain.txt

The screenshot shows a terminal window titled "Activities" with a "Text Editor" tab selected. The terminal window displays the command "tr 'vgapnbrtmosicuxejhqyzflkdw' 'ABCDEFGHIJKLMNPQRSTUVWXYZ' < ciphertext.txt > plain.txt" and its output. The output is a decoded text file named "plain.txt" containing several numbered paragraphs. The text discusses the emergence of MeToo politics, the Golden Globes ceremony, and the Oscar ceremony, mentioning figures like Oprah Winfrey, Natalie Portman, and Catt Sadler. It also mentions the #MeToo movement and the fight against sexual harassment.

```
[10/26/24]seed@VM:~/.../Files$ tr 'vgapnbrtmosicuxejhqyzflkdw' 'ABCDEFGHIJKLMNPQRSTUVWXYZ' < ciphertext.txt > plain.txt
[10/26/24]seed@VM:~/.../Files$
```

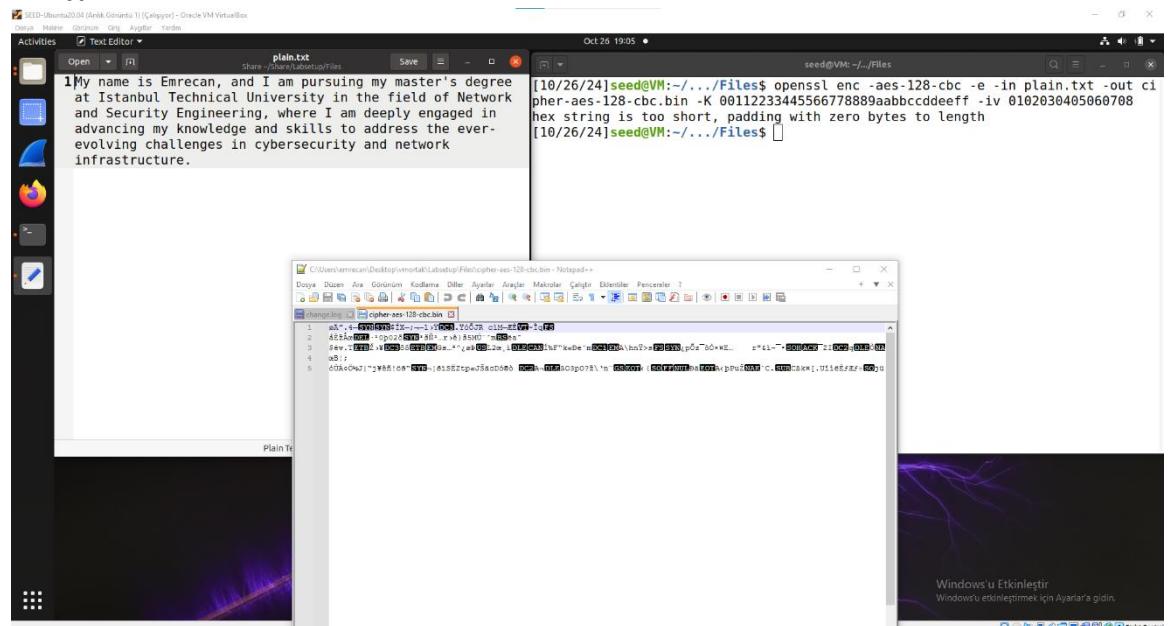
*plain.txt

```
6 THE EMERGENCE OF METOO TIMES UP BLACKGOWN POLITICS ARMANDY ACTIVISM AND
7 A NATIONAL CONVERSATION AS BRIEF AND MAD AS A FEVER DREAM ABOUT WHETHER THERE
8 OUGHT TO BE A PRESIDENT WINFREY THE SEASON DIDNT JUST SEEM EXTRA LONG IT WAS
9 EXTRA LONG BECAUSE THE OSCARS WERE MOVED TO THE FIRST WEEKEND IN MARCH TO
10 AVOID CONFLICTING WITH THE CLOSING CEREMONY OF THE WINTER OLYMPICS THANKS
11 PYEONGCHANG
12
13 ONE BIG QUESTION SURROUNDING THIS YEARS ACADEMY AWARDS IS HOW OR IF THE
14 CEREMONY WILL ADDRESS METOO ESPECIALLY AFTER THE GOLDEN GLOBES WHICH BECAME
15 A JUBILANT COMINGOUT PARTY FOR TIMES UP THE MOVEMENT SPEARHEADED BY
16 POWERFUL HOLLYWOOD WOMEN WHO HELPED RAISE MILLIONS OF DOLLARS TO FIGHT SEXUAL
17 HARASSMENT AROUND THE COUNTRY
18
19 SIGNALING THEIR SUPPORT GOLDEN GLOBES ATTENDEES SWATHED THEMSELVES IN BLACK
20 SPOTTED LAPEL PINS AND SOUNDED OFF ABOUT SEXIST POWER IMBALANCES FROM THE RED
21 CARPET AND THE STAGE ON THE AIR E WAS CALLED OUT ABOUT PAY INEQUITY AFTER
22 ITS FORMER ANCHOR CATT SADLER QUIT ONCE SHE LEARNED THAT SHE WAS MAKING FAR
23 LESS THAN A MALE COHOST AND DURING THE CEREMONY NATALIE PORTMAN TOOK A BLUNT
24 AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOMINATED DIRECTORS HOW COULD
25 THAT BE TOPPED
26
27 AS IT TURNS OUT AT LEAST IN TERMS OF THE OSCARS IT PROBABLY WONT BE
28
29 WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SIGNIFIED THE
30 INITIATIVES LAUNCH THEY NEVER INTENDED IT TO BE JUST AN AWARDS SEASON
31 CAMPAIGN OR ONE THAT BECAME ASSOCIATED ONLY WITH REDCARPET ACTIONS INSTEAD
32 A SPOKESWOMAN SAID THE GROUP IS WORKING BEHIND CLOSED DOORS AND HAS SINCE
33 AMASSED MILLION FOR ITS LEGAL DEFENSE FUND WHICH AFTER THE GLOBES WAS
34 FLOODED WITH THOUSANDS OF DONATIONS OF OR LESS FROM PEOPLE IN SOME
35 COUNTRIES
```

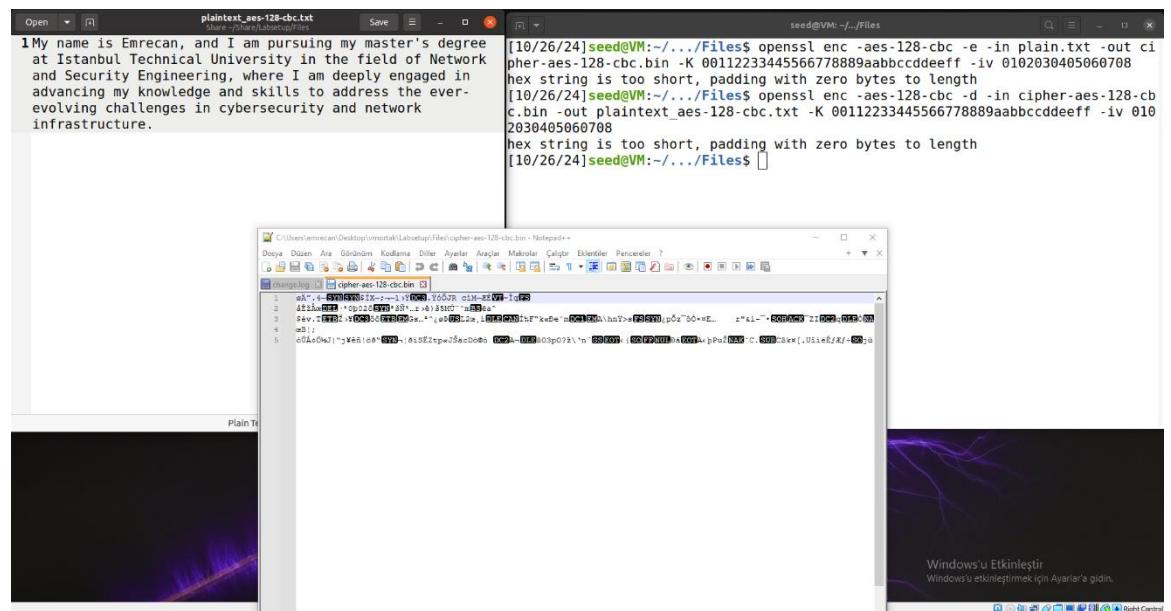
Windows'u Etkinleştir
Windows'u etkinleştirme için Ayarlar'a gidin.

Task 2: Encryption using Different Ciphers and Modes

Encryption AES-128-CBC



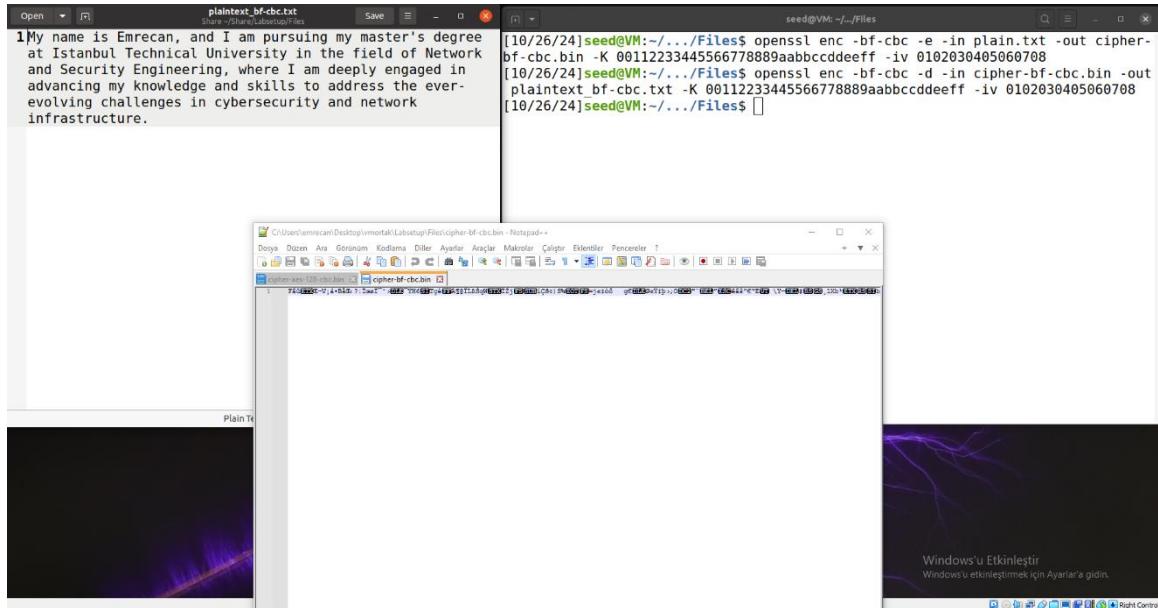
```
openssl enc -aes-128-cbc -e -in plain.txt -out cipher-aes-128-cbc.bin -K 00112233445566778899aabbccddeeff -iv 0102030405060708
```



```
openssl enc -aes-128-cbc -d -in cipher-aes-128-cbc.bin -out plaintext_aes-128-cbc.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
```

Encryption BF-CBC

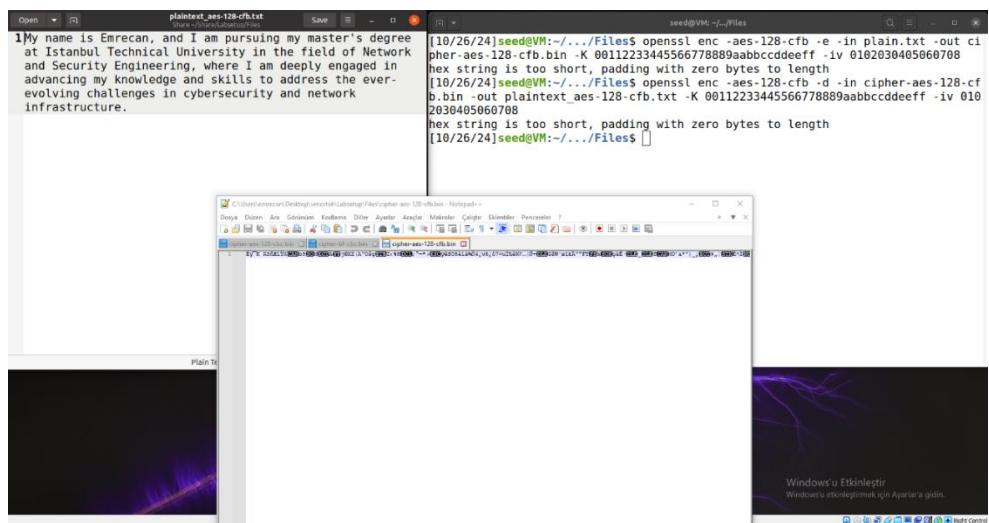
```
openssl enc -bf-cbc -e -in plain.txt -out cipher-bf-cbc.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708
```



```
openssl enc -bf-cbc -d -in cipher-bf-cbc.bin -out plaintext_bf-cbc.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
```

Encryption AES-128-CFB

```
openssl enc -aes-128-cfb -e -in plain.txt -out cipher-aes-128-cfb.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708
```



```
openssl enc -aes-128-cfb -d -in cipher-aes-128-cfb.bin -out plaintext_aes-128-cfb.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
```

AES-128-CBC and AES-256-CBC

Every block is encrypted using the data from the previous encrypted block in a chain-like fashion. This mode creates a connection between data chunks and consistently generates distinct output. High error propagation results from a mistake in one block that impacts the decryption of all blocks that follow. This setting removes patterns, improving data privacy.

AES-128-CFB and AES-256-CFB

It is independent of block size and is a stream cipher mode. Real-time data stream encryption is made possible by its ability to function at the byte or bit level rather than data blocks. only impacts the corrupted bit or byte's decryption.

BF-CBC-128 and BF-CBC-256

An alternative to AES, it is an earlier block cipher algorithm. Despite being flexible and safe, it can run more slowly, particularly when dealing with big datasets.

Task 3: Encryption Mode – ECB vs. CBC

Original Picture:



ECB Mode Analysis:

The screenshot shows a terminal window on a Linux desktop environment. The terminal output is as follows:

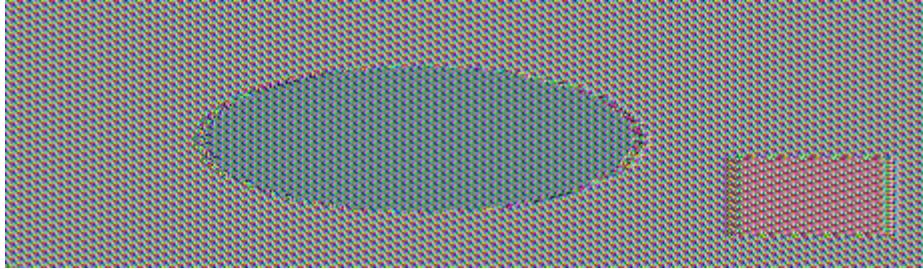
```
[10/26/24]seed@VM:~/.../Files$ openssl enc -aes-128-ecb -in pic_original.bmp -out enc-128-ecb.bmp
enter aes-128-ecb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[10/26/24]seed@VM:~/.../Files$ head -c 54 pic_original.bmp > header
[10/26/24]seed@VM:~/.../Files$ tail -c +55 enc-128-ecb.bmp > body
[10/26/24]seed@VM:~/.../Files$ cat header body > enc-128-ecb.bmp
[10/26/24]seed@VM:~/.../Files$ openssl enc -aes-256-ecb -in pic_original.bmp -out enc-256-ecb.bmp
enter aes-256-ecb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[10/26/24]seed@VM:~/.../Files$ head -c 54 pic_original.bmp > header
[10/26/24]seed@VM:~/.../Files$ tail -c +55 enc-256-ecb.bmp > body
[10/26/24]seed@VM:~/.../Files$ cat header body > enc-256-ecb.bmp
[10/26/24]seed@VM:~/.../Files$
```

Commands:

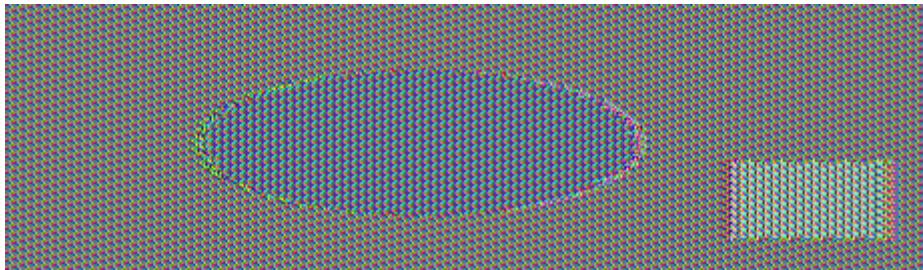
```
openssl enc -aes-128-ecb -in pic_original.bmp -out enc-128-ecb.bmp
head -c 54 pic_original.bmp > header
tail -c +55 enc-128-ecb.bmp > body
cat header body > enc-128-ecb.bmp

openssl enc -aes-256-ecb -in pic_original.bmp -out enc-256-ecb.bmp
head -c 54 pic_original.bmp > header
tail -c +55 enc-256-ecb.bmp > body
cat header body > enc-256-ecb.bmp
```

Output:



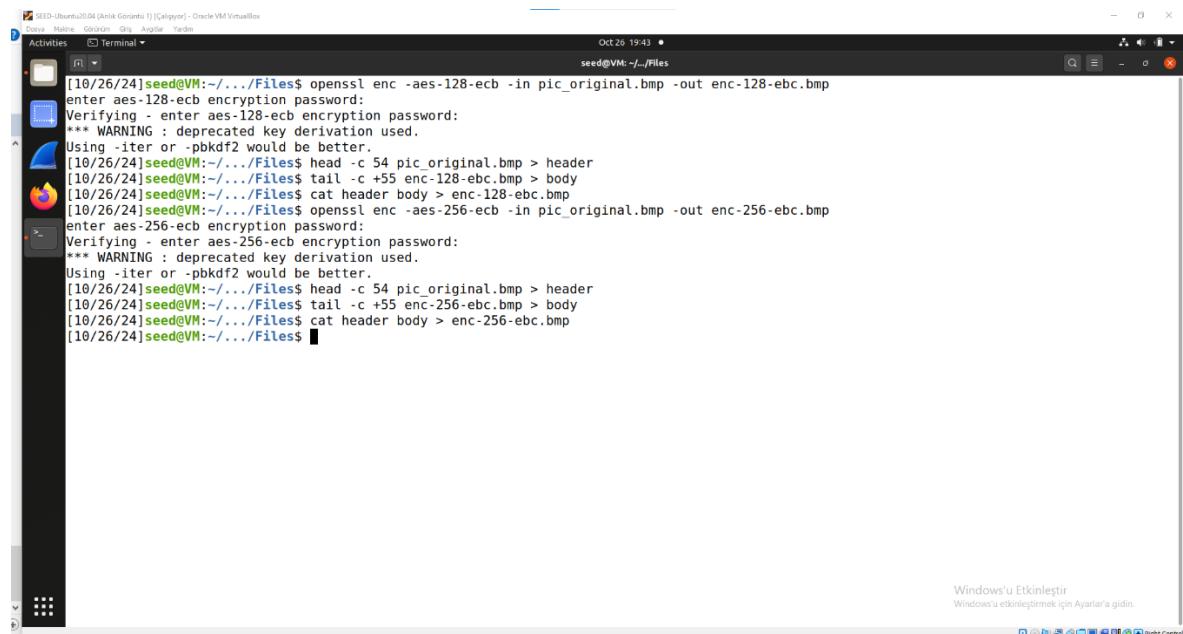
ECB-128



ECB-256

There are visible variations between AES-128 and AES-256. Because the method uses a bigger key space, more diverse color changes and tones can be seen when encrypting the same data.

CBC Mode Analysis:



```
[10/26/24]seed@VM:~/.../Files$ openssl enc -aes-128-cbc -in pic_original.bmp -out enc-128-cbc.bmp
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[10/26/24]seed@VM:~/.../Files$ head -c 54 pic_original.bmp > header
[10/26/24]seed@VM:~/.../Files$ tail -c +55 enc-128-cbc.bmp > body
[10/26/24]seed@VM:~/.../Files$ cat header body > enc-128-cbc.bmp
[10/26/24]seed@VM:~/.../Files$ openssl enc -aes-256-cbc -in pic_original.bmp -out enc-256-cbc.bmp
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.
[10/26/24]seed@VM:~/.../Files$ head -c 54 pic_original.bmp > header
[10/26/24]seed@VM:~/.../Files$ tail -c +55 enc-256-cbc.bmp > body
[10/26/24]seed@VM:~/.../Files$ cat header body > enc-256-cbc.bmp
[10/26/24]seed@VM:~/.../Files$
```

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

Commands:

```
openssl enc -aes-128-cbc -in pic_original.bmp -out enc-128-cbc.bmp
```

```
head -c 54 pic_original.bmp > header
```

```
tail -c +55 enc-128-cbc.bmp > body
```

```
cat header body > enc-128-cbc.bmp
```

```
openssl enc -aes-256-cbc -in pic_original.bmp -out enc-256-cbc.bmp
```

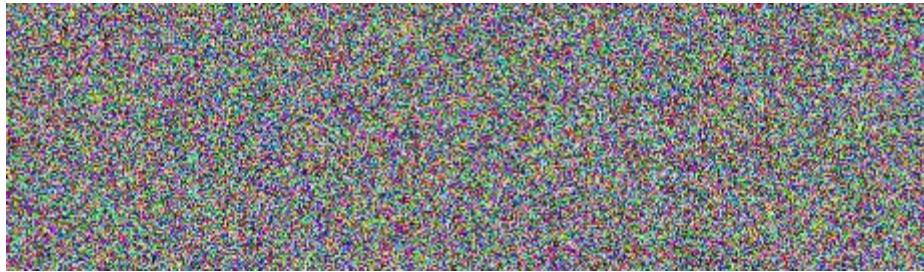
```
head -c 54 pic_original.bmp > header
```

```
tail -c +55 enc-256-cbc.bmp > body
```

```
cat header body > enc-256-cbc.bmp
```

Output:

CBC-128



CBC-256

We can see visible changes between AES-128 and AES-256. Even identical plaintext blocks change into distinct ciphertext blocks in CBC mode because each block is encrypted in a chain with the one before it. AES-256 differs from AES-128 primarily in that it uses a longer key to boost security. There isn't a discernible visual difference in the result, though. In conclusion, the ECB mode is more susceptible to ciphertext analysis due to the patterns it preserves. On the other hand, because data blocks are interdependent, the CBC mode offers a more secure result by hiding patterns and encrypting each block in a unique way. Whereas in CBC mode, errors propagate to succeeding blocks, increasing the level of corruption in the decryption process, in ECB mode, an error in one block only affects that block.

Task 4: Padding

I intended to figure out the messages on my own while using this task. Start by creating 3 files which contain 4 bytes, 8 bytes, and 16 bytes, respectively. Data is encrypted using block cipher techniques in numerous block sizes. The remaining space is filled with padding if the data is not a full block.

```
echo -n "emre" > plain-1.txt == 4 byte  
echo -n "emrecant" > plain-2.txt == 8 byte  
echo -n "emrecantuzeremre" > plain-3.txt == 16 byte
```

Padding AES-128-CBC AES-256-CBC

```
openssl enc -aes-128-cbc -e -in plain-1.txt -out enc-cipher-1-cbc.bin  
openssl enc -aes-128-cbc -e -in plain-2.txt -out enc-cipher-2-cbc.bin  
openssl enc -aes-128-cbc -e -in plain-3.txt -out enc-cipher-3-cbc.bin  
openssl enc -aes-128-cbc -d -nopad -in enc-cipher-1-cbc.bin -out dec-plain-1-cbc.txt  
openssl enc -aes-128-cbc -d -nopad -in enc-cipher-2-cbc.bin -out dec-plain-2-cbc.txt  
openssl enc -aes-128-cbc -d -nopad -in enc-cipher-3-cbc.bin -out dec-plain-3-cbc.txt
```



The screenshot shows a terminal window titled 'Activities' with a green title bar. The window contains a command-line interface where the user is performing a series of operations:

- Encryption of 'plain-1.txt' into 'enc-cipher-1-cbc.bin' using AES-128-CBC encryption.
- Encryption of 'plain-2.txt' into 'enc-cipher-2-cbc.bin' using AES-128-CBC encryption.
- Encryption of 'plain-3.txt' into 'enc-cipher-3-cbc.bin' using AES-128-CBC encryption.
- Decryption of 'enc-cipher-1-cbc.bin' back into 'dec-plain-1-cbc.txt' using AES-128-CBC decryption.
- Decryption of 'enc-cipher-2-cbc.bin' back into 'dec-plain-2-cbc.txt' using AES-128-CBC decryption.
- Decryption of 'enc-cipher-3-cbc.bin' back into 'dec-plain-3-cbc.txt' using AES-128-CBC decryption.

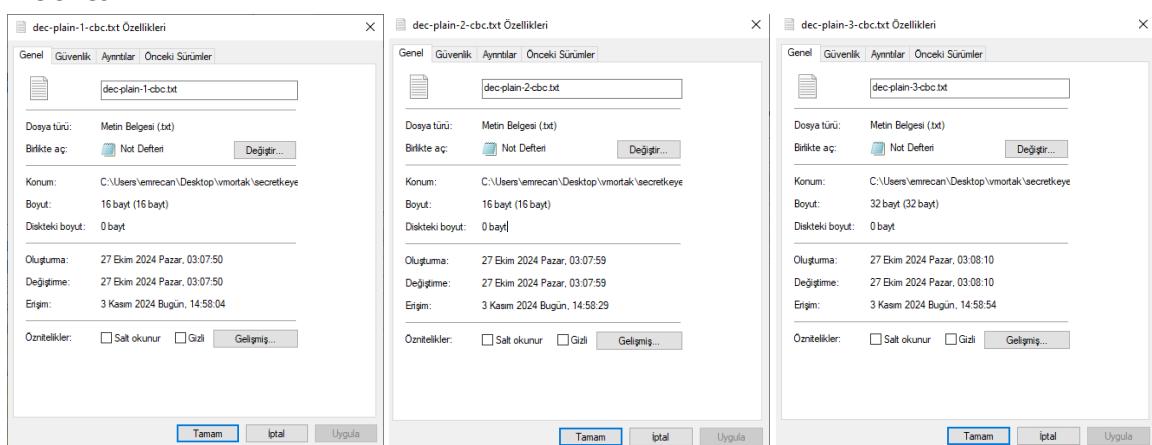
The terminal output includes several 'WARNING' messages about deprecated key derivation methods like 'pbkdf2' and 'iter'.

```

[10/26/24]seed@VM:~/.../Files$ cd task
task1/ task2/ task3/ task4/
[10/26/24]seed@VM:~/.../Files$ cd task4/
[10/26/24]seed@VM:~/.../task4$ cd cbc128/
[10/26/24]seed@VM:~/.../cbc128$ xxd
dec-plain-1-cbc.txt dec-plain-2-cbc.txt dec-plain-3-cbc.txt enc-cipher-1-cbc.bin enc-cipher-2-cbc.bin enc-cipher-3-cbc.bin
[10/26/24]seed@VM:~/.../cbc128$ xxd dec-plain-1-cbc.txt
00000000: 656d 7265 0c0c 0c0c 0c0c 0c0c emre.....
[10/26/24]seed@VM:~/.../cbc128$ xxd dec-plain-2-cbc.txt
00000000: 656d 7265 6361 6e74 0808 0808 0808 emrecant.....
[10/26/24]seed@VM:~/.../cbc128$ xxd dec-plain-3-cbc.txt
00000000: 656d 7265 6361 6e74 757a 6572 656d 7265 emrecantuzeremre
00000010: 1010 1010 1010 1010 1010 1010 1010 1010 .....
[10/26/24]seed@VM:~/.../cbc128$ 

```

File Sizes:

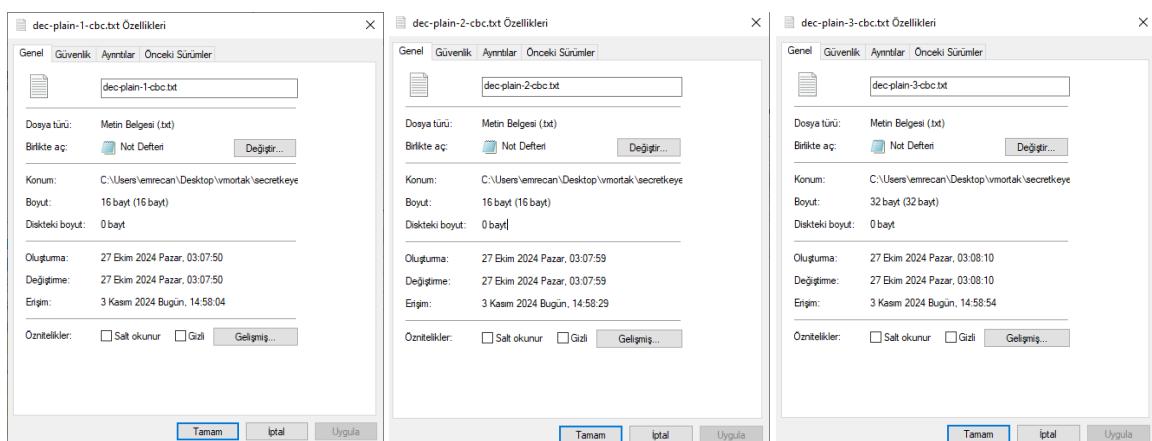


Prior to encryption, the file size was 4,8,16 bytes. The data is padded with 12,8,16 bytes to fill the block size. File sizes rise to 16,16,32 bytes after encryption. The padding is eliminated during decryption, restoring the original size. The file size is 16 bytes, or the entire block size, prior to encryption. The file size grows to 32 bytes as a full block is filled, adding an additional 16 bytes of padding.

```

[18/26/24]seed@VM:~/.../Files$ openssl enc -aes-256-cbc -e -in plain-1.txt -out enc-cipher-1-cbc.bin
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.
[18/26/24]seed@VM:~/.../Files$ openssl enc -aes-256-cbc -e -in plain-2.txt -out enc-cipher-2-cbc.bin
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.
[18/26/24]seed@VM:~/.../Files$ openssl enc -aes-256-cbc -e -in plain-3.txt -out enc-cipher-3-cbc.bin
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.
[18/26/24]seed@VM:~/.../Files$ openssl enc -aes-256-cbc -d -nopad -in enc-cipher-1-cbc.bin -out dec-plain-1-cbc.txt
enter aes-256-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[18/26/24]seed@VM:~/.../Files$ openssl enc -aes-256-cbc -d -nopad -in enc-cipher-2-cbc.bin -out dec-plain-2-cbc.txt
enter aes-256-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[18/26/24]seed@VM:~/.../Files$ openssl enc -aes-256-cbc -d -nopad -in enc-cipher-3-cbc.bin -out dec-plain-3-cbc.txt
enter aes-256-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[18/26/24]seed@VM:~/.../Files$
```

File Sizes:



Prior to encryption, the file size was 4,8,16 bytes. The data is padded with 12,8,16 bytes to fill the block size. File sizes rise to 16,16,32 bytes after encryption. The padding is eliminated during decryption, restoring the original size. The file size is 16 bytes, or the entire block size, prior to encryption. The file size grows to 32 bytes as a full block is filled, adding an additional 16 bytes of padding.

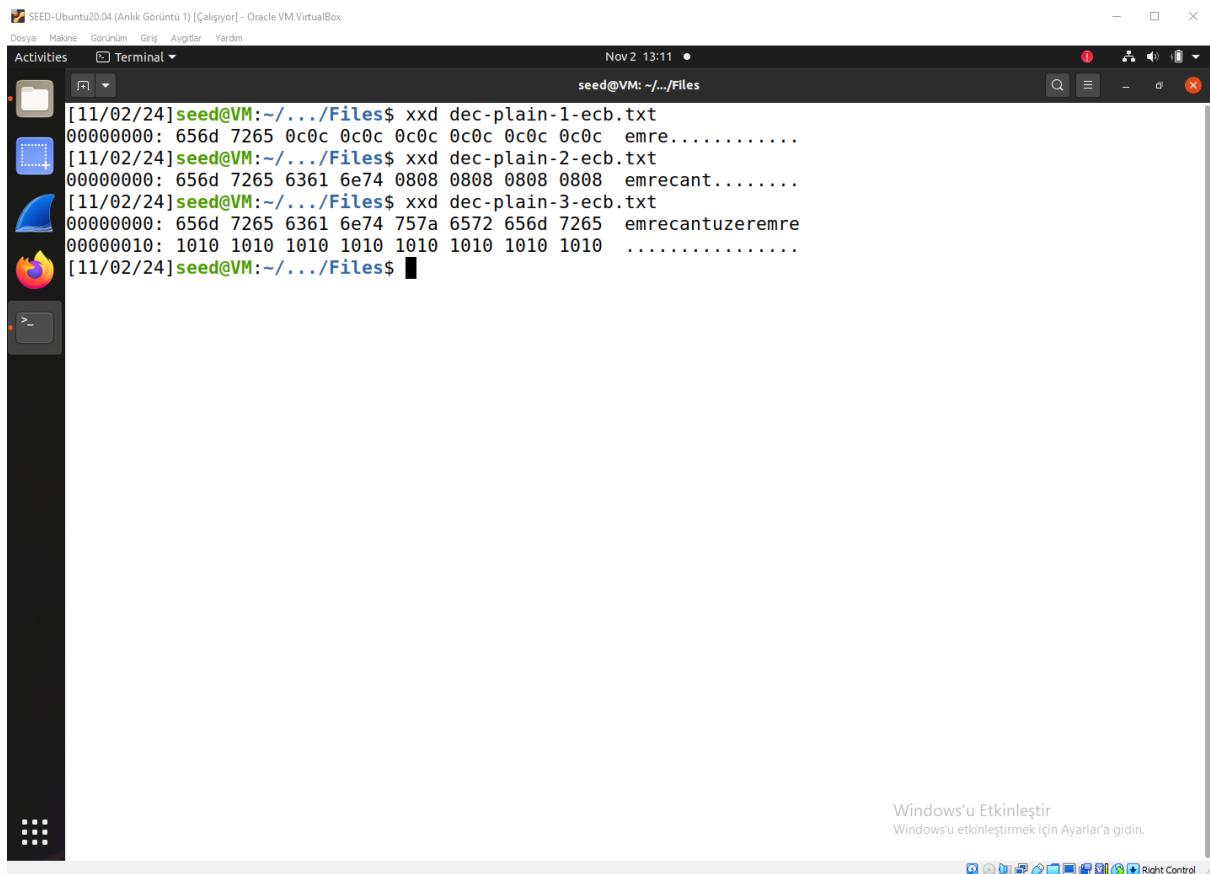
Padding AES-128-ECB AES-256-ECB

```
openssl enc -aes-128-ecb -e -in plain-1.txt -out enc-cipher-1-ecb.bin  
openssl enc -aes-128-ecb -e -in plain-2.txt -out enc-cipher-2-ecb.bin  
openssl enc -aes-128-ecb -e -in plain-3.txt -out enc-cipher-3-ecb.bin  
openssl enc -aes-128-ecb -d -nopad -in enc-cipher-1-ecb.bin -out dec-plain-1-ecb.txt  
openssl enc -aes-128-ecb -d -nopad -in enc-cipher-2-ecb.bin -out dec-plain-2-ecb.txt  
openssl enc -aes-128-ecb -d -nopad -in enc-cipher-3-ecb.bin -out dec-plain-3-ecb.txt
```

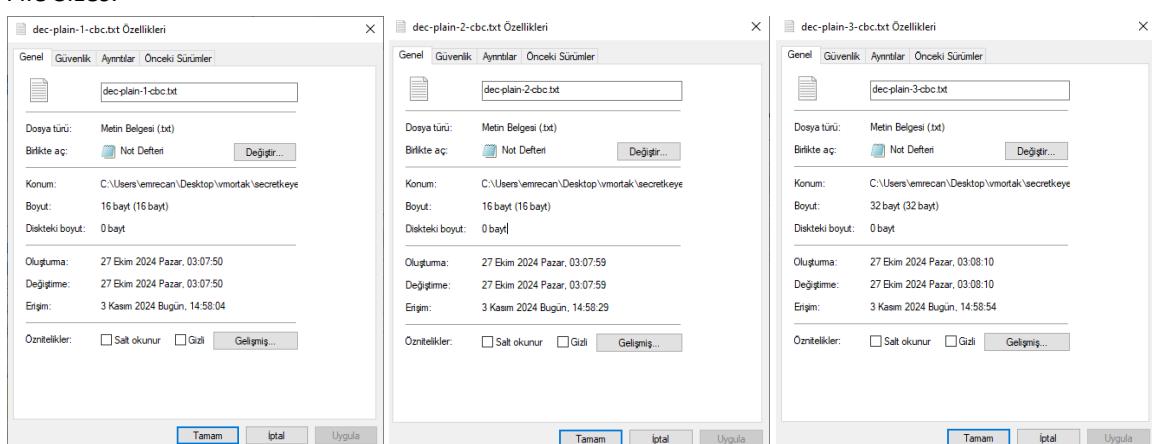


The screenshot shows a terminal window titled "seed@VM: ~/.Files\$" running on a Linux desktop environment. The terminal displays the command-line interface for OpenSSL, specifically for performing encryption and decryption operations using the AES-128-ECB cipher mode. The user enters commands to encrypt three files ("plain-1.txt", "plain-2.txt", and "plain-3.txt") into binary files ("enc-cipher-1-ecb.bin", "enc-cipher-2-ecb.bin", and "enc-cipher-3-ecb.bin"). The user also performs decryption operations using the "-d" option and the "-nopad" flag to produce the original plain text files ("dec-plain-1-ecb.txt", "dec-plain-2-ecb.txt", and "dec-plain-3-ecb.txt"). The terminal output includes several "WARNING" messages about deprecated key derivation methods like "pbkdf2". The desktop environment includes a dock with icons for Home, Dash, Activities, Games, and Terminal, and a status bar at the bottom.

```
[11/02/24]seed@VM:~/.Files$ openssl enc -aes-128-ecb -e -in plain-1.txt -out enc-cipher-1-ecb.bin  
enter aes-128-ecb encryption password:  
Verifying - enter aes-128-ecb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.Files$ openssl enc -aes-128-ecb -e -in plain-2.txt -out enc-cipher-2-ecb.bin  
enter aes-128-ecb encryption password:  
Verifying - enter aes-128-ecb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.Files$ openssl enc -aes-128-ecb -e -in plain-3.txt -out enc-cipher-3-ecb.bin  
enter aes-128-ecb encryption password:  
Verifying - enter aes-128-ecb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.Files$ openssl enc -aes-128-ecb -d -nopad -in enc-cipher-1-ecb.bin -out dec-plain-1-ecb.txt  
enter aes-128-ecb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.Files$ openssl enc -aes-128-ecb -d -nopad -in enc-cipher-2-ecb.bin -out dec-plain-2-ecb.txt  
enter aes-128-ecb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.Files$ openssl enc -aes-128-ecb -d -nopad -in enc-cipher-3-ecb.bin -out dec-plain-3-ecb.txt  
enter aes-128-ecb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.Files$
```



File Sizes:

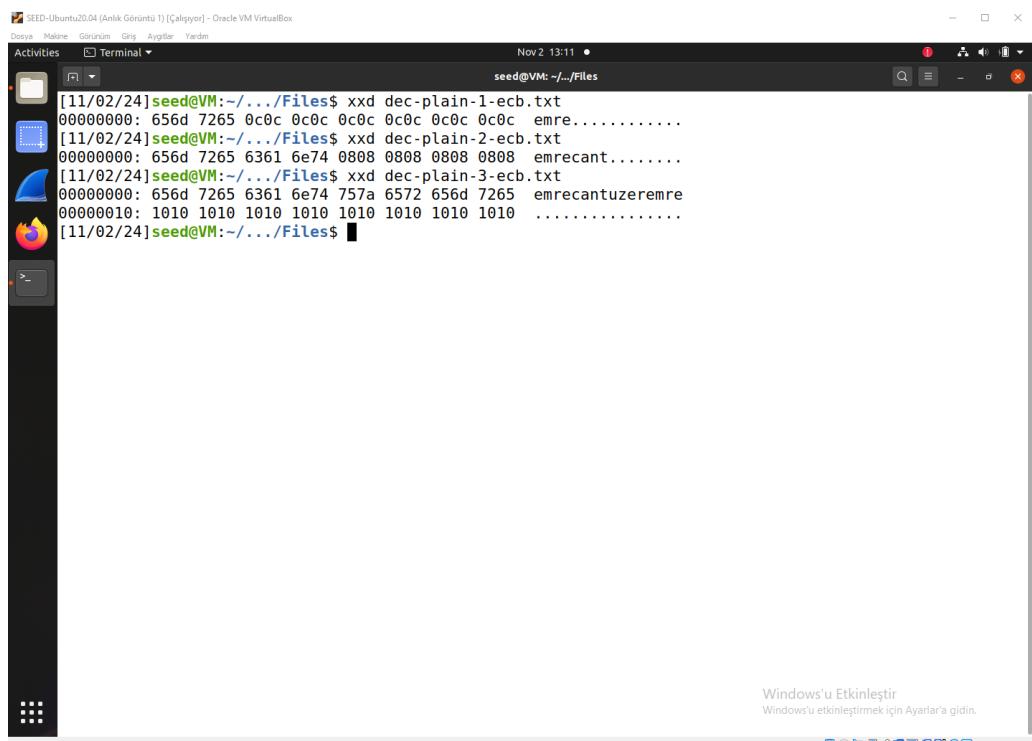


Prior to encryption, the file size was 4,8,16 bytes. The data is padded with 12,8,16 bytes to fill the block size. File sizes rise to 16,16,32 bytes after encryption. The padding is eliminated during decryption, restoring the original size. The file size is 16 bytes, or the entire block size, prior to encryption. The file size grows to 32 bytes as a full block is filled, adding an additional 16 bytes of padding.

```
openssl enc -aes-256-ecb -e -in plain-1.txt -out enc-cipher-1-ecb.bin  
openssl enc -aes-256-ecb -e -in plain-2.txt -out enc-cipher-2-ecb.bin  
openssl enc -aes-256-ecb -e -in plain-3.txt -out enc-cipher-3-ecb.bin  
openssl enc -aes-256-ecb -d -nopad -in enc-cipher-1-ecb.bin -out dec-plain-1-ecb.txt  
openssl enc -aes-256-ecb -d -nopad -in enc-cipher-2-ecb.bin -out dec-plain-2-ecb.txt  
openssl enc -aes-256-ecb -d -nopad -in enc-cipher-3-ecb.bin -out dec-plain-3-ecb.txt
```

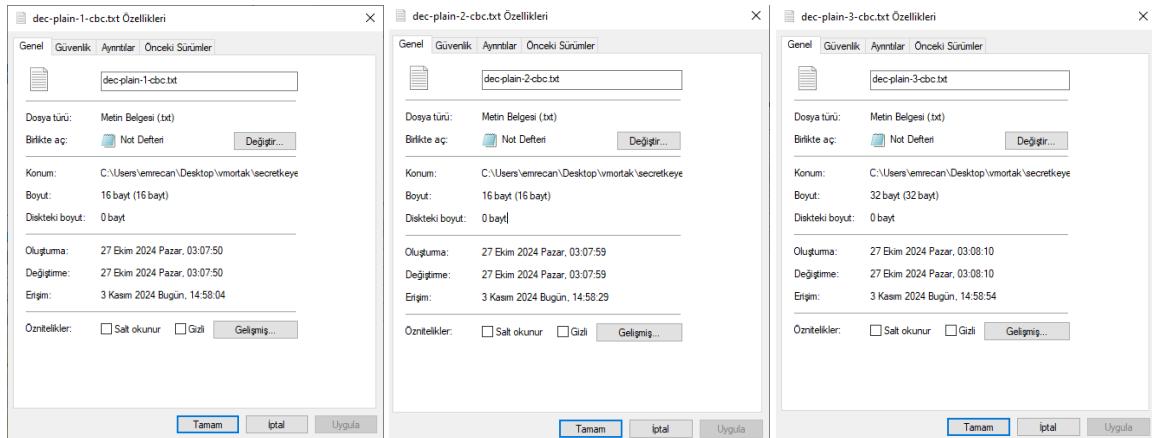


```
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-256-ecb -e -in plain-1.txt -out enc-cipher-1-ecb.bin  
enter aes-256-ecb encryption password:  
Verifying - enter aes-256-ecb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-256-ecb -e -in plain-2.txt -out enc-cipher-2-ecb.bin  
enter aes-256-ecb encryption password:  
Verifying - enter aes-256-ecb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-256-ecb -e -in plain-3.txt -out enc-cipher-3-ecb.bin  
enter aes-256-ecb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-256-ecb -d -nopad -in enc-cipher-1-ecb.bin -out dec-plain-1-ecb.txt  
enter aes-256-ecb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-256-ecb -d -nopad -in enc-cipher-2-ecb.bin -out dec-plain-2-ecb.txt  
enter aes-256-ecb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-256-ecb -d -nopad -in enc-cipher-3-ecb.bin -out dec-plain-3-ecb.txt  
enter aes-256-ecb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24] seed@VM:~/.../Files$
```



```
[11/02/24] seed@VM:~/.../Files$ xxd dec-plain-1-ecb.txt  
00000000: 656d 7265 0c0c 0c0c 0c0c 0c0c emre.....  
[11/02/24] seed@VM:~/.../Files$ xxd dec-plain-2-ecb.txt  
00000000: 656d 7265 6361 6e74 0808 0808 0808 emrecant.....  
[11/02/24] seed@VM:~/.../Files$ xxd dec-plain-3-ecb.txt  
00000000: 656d 7265 6361 6e74 757a 6572 656d 7265 emrecantuzeremre  
00000010: 1010 1010 1010 1010 1010 1010 1010 1010 .....  
[11/02/24] seed@VM:~/.../Files$
```

File Sizes:



Prior to encryption, the file size was 4,8,16 bytes. The data is padded with 12,8,16 bytes to fill the block size. File sizes rise to 16,16,32 bytes after encryption. The padding is eliminated during decryption, restoring the original size. The file size is 16 bytes, or the entire block size, prior to encryption. The file size grows to 32 bytes as a full block is filled, adding an additional 16 bytes of padding.

Padding AES-128-CFB AES-256-CFB

```
openssl enc -aes-128-cfb -e -in plain-1.txt -out enc-cipher-1-cfb.bin
openssl enc -aes-128-cfb -e -in plain-2.txt -out enc-cipher-2-cfb.bin
openssl enc -aes-128-cfb -e -in plain-3.txt -out enc-cipher-3-cfb.bin
openssl enc -aes-128-cfb -d -nopad -in enc-cipher-1-cfb.bin -out dec-plain-1-cfb.txt
openssl enc -aes-128-cfb -d -nopad -in enc-cipher-2-cfb.bin -out dec-plain-2-cfb.txt
openssl enc -aes-128-cfb -d -nopad -in enc-cipher-3-cfb.bin -out dec-plain-3-cfb.txt
openssl enc -aes-256-cfb -e -in plain-1.txt -out enc-cipher-1-cfb.bin
openssl enc -aes-256-cfb -e -in plain-2.txt -out enc-cipher-2-cfb.bin
openssl enc -aes-256-cfb -e -in plain-3.txt -out enc-cipher-3-cfb.bin
openssl enc -aes-256-cfb -d -nopad -in enc-cipher-1-cfb.bin -out dec-plain-1-cfb.txt
openssl enc -aes-256-cfb -d -nopad -in enc-cipher-2-cfb.bin -out dec-plain-2-cfb.txt
openssl enc -aes-256-cfb -d -nopad -in enc-cipher-3-cfb.bin -out dec-plain-3-cfb.txt
```

```

[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-256-cfb -e -in plain-1.txt -out enc-cipher-1-cfb.bin
enter aes-256-cfb encryption password:
Verifying - enter aes-256-cfb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-256-cfb -e -in plain-2.txt -out enc-cipher-2-cfb.bin
enter aes-256-cfb encryption password:
Verifying - enter aes-256-cfb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-256-cfb -e -in plain-3.txt -out enc-cipher-3-cfb.bin
enter aes-256-cfb encryption password:
Verifying - enter aes-256-cfb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-256-cfb -d -nopad -in enc-cipher-1-cfb.bin -out dec-plain-1-cfb.txt
enter aes-256-cfb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-256-cfb -d -nopad -in enc-cipher-2-cfb.bin -out dec-plain-2-cfb.txt
enter aes-256-cfb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-256-cfb -d -nopad -in enc-cipher-3-cfb.bin -out dec-plain-3-cfb.txt
enter aes-256-cfb decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ SS

```

Windows U Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

Fайл	Свойства	Свойства	Свойства
dec-plain-1-cfb.txt	Файл	Файл	Файл
dec-plain-2-cfb.txt	Файл	Файл	Файл
dec-plain-3-cfb.txt	Файл	Файл	Файл

In streaming modes, the file size doesn't change while encryption and decryption are underway. Before and after encryption and decryption, the files' sizes don't change. These modes function independently of the data stream's block sizes.

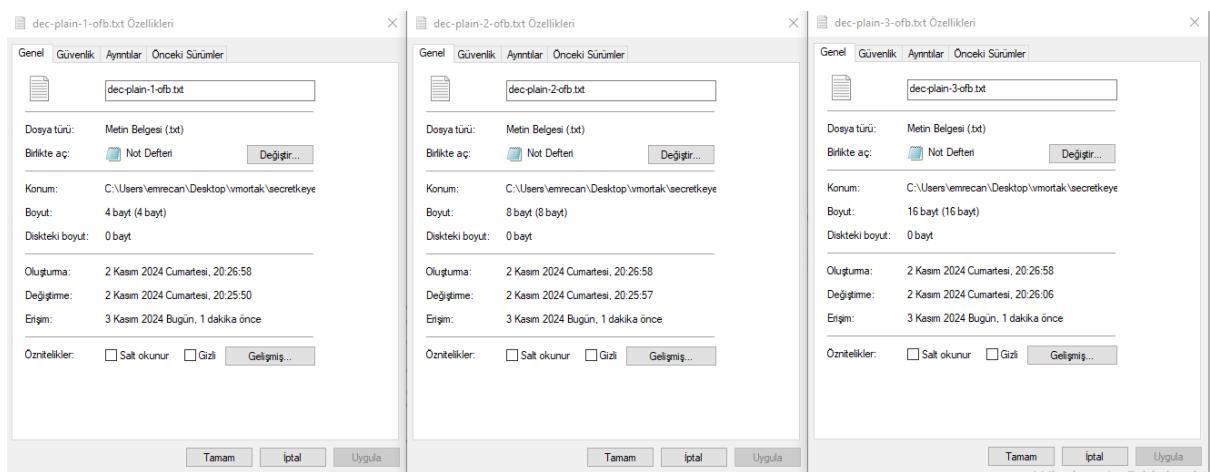
Padding AES-128-CFB AES-256-CFB

```
openssl enc -aes-128-ofb -e -in plain-1.txt -out enc-cipher-1-ofb.bin  
openssl enc -aes-128-ofb -e -in plain-2.txt -out enc-cipher-2-ofb.bin  
openssl enc -aes-128-ofb -e -in plain-3.txt -out enc-cipher-3-ofb.bin  
openssl enc -aes-128-ofb -d -nopad -in enc-cipher-1-ofb.bin -out dec-plain-1-cfb.txt  
openssl enc -aes-128-ofb -d -nopad -in enc-cipher-2-ofb.bin -out dec-plain-2-cfb.txt  
openssl enc -aes-128-ofb -d -nopad -in enc-cipher-3-ofb.bin -out dec-plain-3-cfb.txt  
openssl enc -aes-256-ofb -e -in plain-1.txt -out enc-cipher-1-cfb.bin  
openssl enc -aes-256-ofb -e -in plain-2.txt -out enc-cipher-2-cfb.bin  
openssl enc -aes-256-ofb -e -in plain-3.txt -out enc-cipher-3-cfb.bin  
openssl enc -aes-256-ofb -d -nopad -in enc-cipher-1-ofb.bin -out dec-plain-1-cfb.txt  
openssl enc -aes-256-ofb -d -nopad -in enc-cipher-2-ofb.bin -out dec-plain-2-cfb.txt  
openssl enc -aes-256-ofb -d -nopad -in enc-cipher-3-ofb.bin -out dec-plain-3-cfb.txt
```



The screenshot shows a terminal window titled "SEED-Ubuntu20.04 (Anlık Görüntü 1) [Çalışıyor] - Oracle VM VirtualBox". The terminal window displays a series of OpenSSL commands being run on a file named "plain-1.txt" through "plain-3.txt". The commands involve encryption and decryption using AES-128 and AES-256 with CFB mode, and they demonstrate the use of different padding options (-e for encryption and -d for decryption) and the -nopad option to avoid padding. The terminal also shows the password entry process for each encryption command. The output includes several "WARNING : deprecated key derivation used." messages.

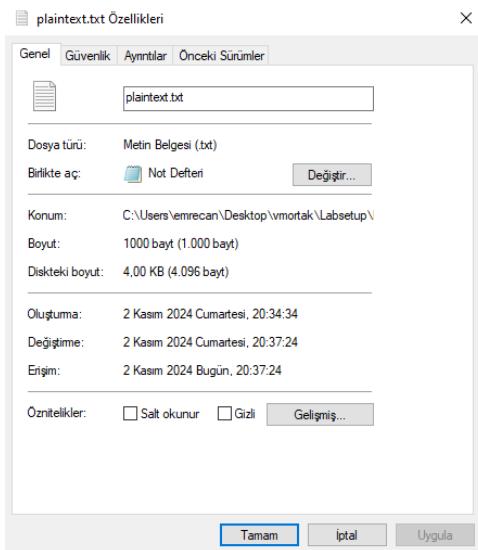
```
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-128-ofb -e -in plain-1.txt -out enc-cipher-1-cfb.bin  
enter aes-128-ofb encryption password:  
Verifying - enter aes-128-ofb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-128-ofb -e -in plain-2.txt -out enc-cipher-2-cfb.bin  
enter aes-128-ofb encryption password:  
Verifying - enter aes-128-ofb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-128-ofb -e -in plain-3.txt -out enc-cipher-3-cfb.bin  
enter aes-128-ofb encryption password:  
Verifying - enter aes-128-ofb encryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-128-ofb -d -nopad -in enc-cipher-1-cfb.bin -out dec-plain-1-cfb.txt  
enter aes-128-ofb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-128-ofb -d -nopad -in enc-cipher-2-cfb.bin -out dec-plain-2-cfb.txt  
enter aes-128-ofb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-128-ofb -d -nopad -in enc-cipher-3-cfb.bin -out dec-plain-3-cfb.txt  
enter aes-128-ofb decryption password:  
*** WARNING : deprecated key derivation used.  
Using -iter or -pbkdf2 would be better.  
[11/02/24]seed@VM:~/.../Files$
```



In streaming modes, the file size doesn't change while encryption and decryption are underway. Before and after encryption and decryption, the files' sizes don't change. These modes function independently of the data stream's block sizes.

Task 5: Error Propagation – Corrupted Cipher Text

Making a new text file with 1000 bytes is the first stage in this project.



ECB Mode Analysis:

In ECB mode, each block is encrypted separately, therefore a mistake in one block will only result in that block's incorrect decryption. This makes it possible to decrypt blocks other than corrupted ones.

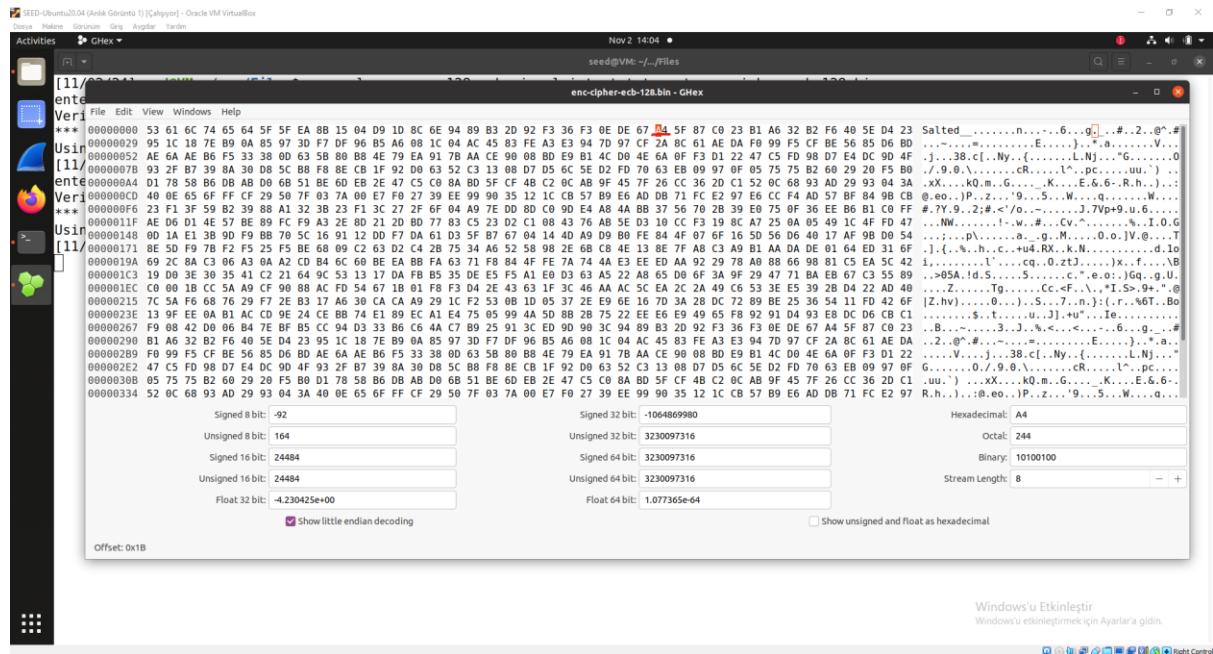
Encryption:

```
openssl enc -aes-128-ecb -in plaintext.txt -out enc-cipher-ecb-128.bin
```

```
openssl enc -aes-256-ecb -in plaintext.txt -out enc-cipher-ecb-256.bin
```

A screenshot of a Linux terminal window titled 'SEED-Ubuntu20.04 (Anlık Görüntü 1) [Çalıpyor] - Oracle VM VirtualBox'. The terminal shows two commands being run: 'openssl enc -aes-128-ecb -in plaintext.txt -out enc-cipher-ecb-128.txt' and 'openssl enc -aes-256-ecb -in plaintext.txt -out enc-cipher-ecb-256.txt'. Both commands prompt for an encryption password and output a warning message: '*** WARNING : deprecated key derivation used.' and 'Using -iter or -pbkdf2 would be better.' The terminal window has a dark theme and is part of the Unity desktop environment.

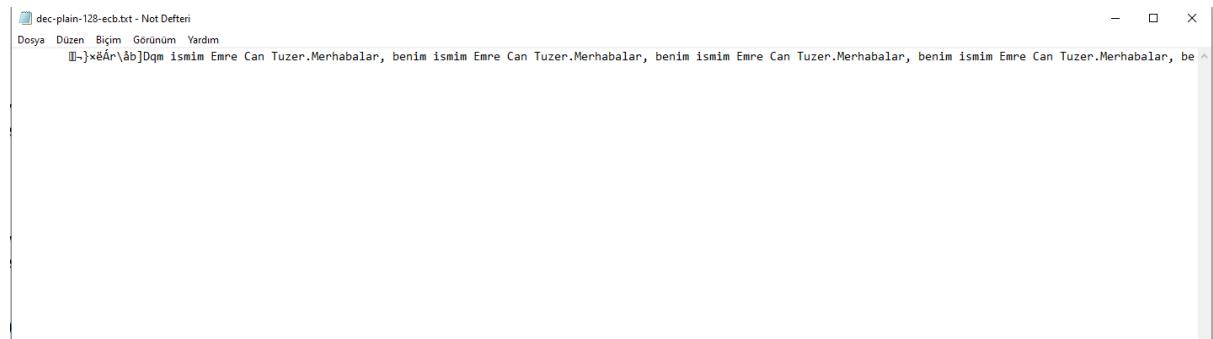
At the 55th Byte, switched from B4 to B5:



Decryption:

```
openssl enc -aes-128-ecb -d -in enc-cipher-ecb-128.bin -out dec-plain-128-ecb.txt
openssl enc -aes-128-ecb -d -in enc-cipher-ecb-256.bin -out dec-plain-256-ecb.txt
```

Output:



The remaining blocks were correctly decoded, with the exception of the faulty block.

CBC Mode Analysis:

When a block in CBC mode becomes corrupted, the decryption of that block fails, and the corrupted block is impacted along with one consecutive block. Because the first block following the bad block is XORed with the output of the bad block that was erroneously decoded, it is inaccurate.

Encryption:

```
openssl enc -aes-128-cbc -in plaintext.txt -out enc-cipher-cbc-128.bin
```

```
openssl enc -aes-256-cbc -in plaintext.txt -out enc-cipher-cbc-256.bin
```

The terminal window shows the following command sequence:

```
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-128-cbc -in plaintext.txt -out enc-cipher-cbc-128.bin
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-256-cbc -in plaintext.txt -out enc-cipher-cbc-256.bin
[11/02/24] seed@VM:~/.../Files$ ghex enc-cipher-cbc-128.bin
[11/02/24] seed@VM:~/.../Files$ ghex enc-cipher-cbc-256.bin
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-128-cbc -d -in enc-cipher-cbc-128.bin -out dec-plain-128-cbc.txt
[11/02/24] seed@VM:~/.../Files$ openssl enc -aes-256-cbc -d -in enc-cipher-cbc-256.bin -out dec-plain-256-cbc.txt
[11/02/24] seed@VM:~/.../Files$ bad decrypt
140624668075328:error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt:crypto/evp/evp_enc.c:583:
[11/02/24] seed@VM:~/.../Files$
```

At the 55th Byte, switched from D6 to D7:

The GHex application displays the hex dump of the encrypted file. A specific byte at offset 55 is highlighted, showing it has been modified from D6 to D7.

Offset: 0x0	Signed 8 bit:	Unsigned 8 bit:	Signed 16 bit:	Unsigned 16 bit:	Signed 32 bit:	Unsigned 32 bit:	Signed 64 bit:	Unsigned 64 bit:	Binary:	Stream Length:
	83	83	24915	24915	1953259859	1953259859	1953259859	1953259859	01010011	8
										- +

Decryption:

```
openssl enc -aes-128-cbc -d -in enc-cipher-cbc-128.bin -out dec-plain-128-cbc.txt
```

```
openssl enc -aes-256-cbc -d -in enc-cipher-cbc-256.bin -out dec-plain-256-cbc.txt
```

Output:

While subsequent blocks were decoded successfully, the block with the 55th byte and the block immediately after it were deciphered wrongly.

CFB Mode Analysis:

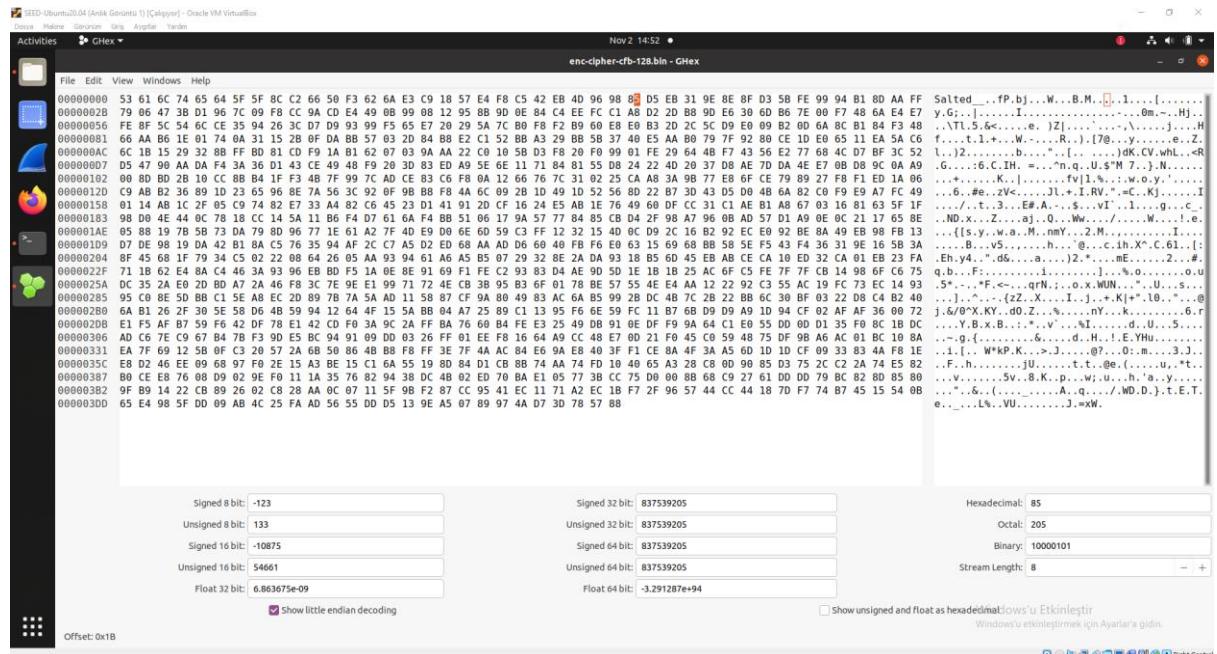
Encryption:

```
openssl enc -aes-128-cfb -in plaintext.txt -out enc-cipher-cfb-128.bin
```

```
openssl enc -aes-256-cfb -in plaintext.txt -out enc-cipher-cfb-256.bin
```

```
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-128-cfb -in plaintext.txt -out enc-cipher-cfb-128.bin
enter aes-128-cfb encryption password:
Verifying - enter aes-128-cfb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-256-cfb -in plaintext.txt -out enc-cipher-cfb-256.bin
enter aes-256-cfb encryption password:
Verifying - enter aes-256-cfb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ ghex enc-cipher-cfb-128.bin
```

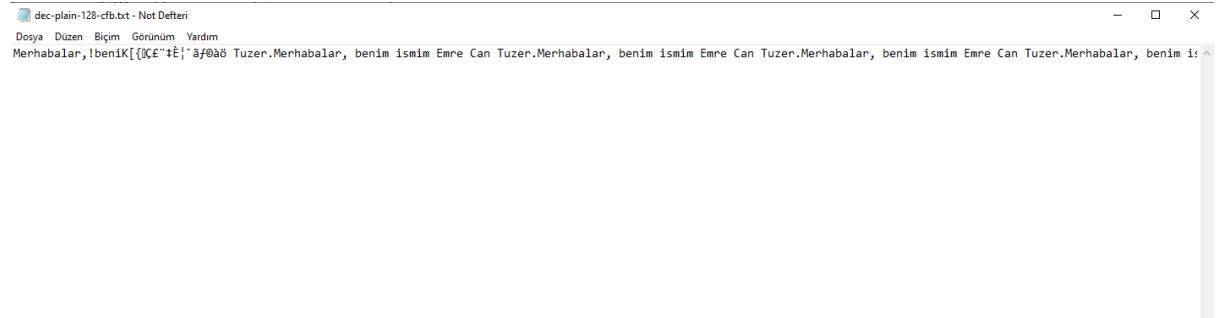
At the 55th Byte, switched from 85 to 86:



Decryption:

```
openssl enc -aes-128-cfb -d -in enc-cipher-cfb-128.bin -out dec-plain-128-cfb.txt  
openssl enc -aes-256-cfb -d -in enc-cipher-cfb-256.bin -out dec-plain-256-cfb.txt
```

Output:



OFB Mode Analysis:

In OFB mode, a string from the encryption stream is separately XORed with each block. Because of this, OFB mode is less likely to propagate errors. Only that block is impacted when it becomes corrupted.

Encryption:

```
openssl enc -aes-128-ofb -in plaintext.txt -out enc-cipher-ofb-128.bin  
openssl enc -aes-256-ofb -in plaintext.txt -out enc-cipher-ofb-256.bin
```

```
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-128-ofb -in plaintext.txt -out enc-cipher-ofb-128.bin
enter aes-128-ofb encryption password:
Verifying - enter aes-128-ofb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ openssl enc -aes-256-ofb -in plaintext.txt -out enc-cipher-ofb-256.bin
enter aes-256-ofb encryption password:
Verifying - enter aes-256-ofb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
[11/02/24]seed@VM:~/.../Files$ ghex enc-cipher-ofb-128.bin
```

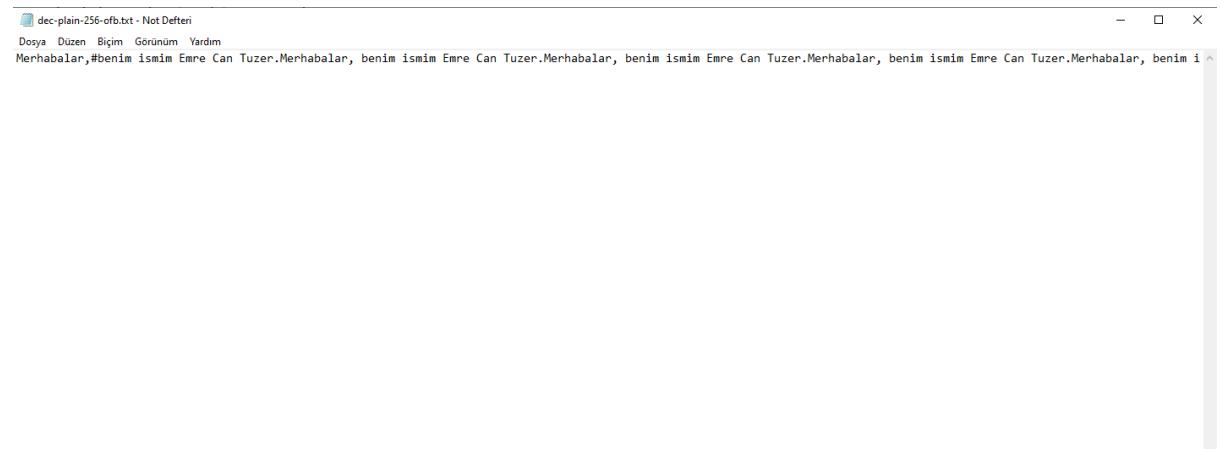
At the 55th Byte, switched from FE to FF:

Signed 8 bit:	1953259859	Hexadecimal:	53
Unsigned 8 bit:	1953259859	Octal:	123
Signed 16 bit:	1953259859	Binary:	01010011
Unsigned 16 bit:	1953259859	Stream Length:	8
Float 32 bit:	7.491187e+31		
<input checked="" type="checkbox"/> Show little endian decoding			

Decryption:

```
openssl enc -aes-128-cfb -d -in enc-cipher-cfb-128.bin -out dec-plain-128-cfb.txt
openssl enc -aes-256-cfb -d -in enc-cipher-cfb-256.bin -out dec-plain-256-cfb.txt
```

Output:



```
dec-plain-256-ofb.txt - Not Defteri
Dosa Düzen Biçim Görünüm Yardım
Merhabalar,#benim ismim Emre Can Tuze.Merhabalar, benim ismim Emre Can Tuze.Merhabalar, benim ismim Emre Can Tuze.Merhabalar, benim i ^
```

The other blocks were all proper, with the exception of the block that included the 55th byte.

Finally, it has been noted that OFB isolates faults and limits error propagation to a minimum, CBC and CFB cause flaws to spread by upsetting the blockchain and impacting data integrity, while ECB limits errors on a block basis and can be decrypted independently.