BIRZEIT UNIVERSITY

Faculty of Engineering & Technology
Electrical & Computer Engineering Department

ENCS4320

Cryptography Lab Report

Prepared by:

Mohammad Al-Sayyed 1180154

Tareq Shannak 1181404

Instructor: Dr. Hanna Al-Zughbi

Section: 2

Date: 31 October 2021

Table of Contents

List of Figures:

Task 1: Frequency Analysis

Step 1

We can see the mono-alphabetic substitution cipher in the figure below.

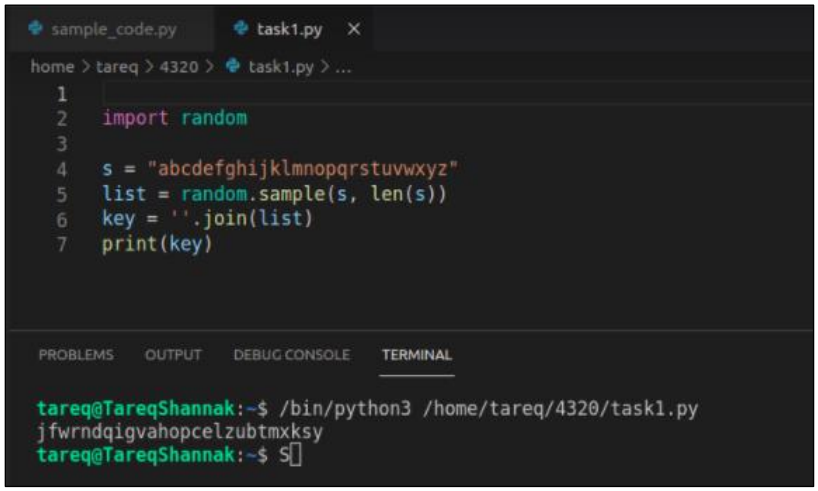


Figure 1 - Generating random key

Step 2

In this figure, the plain text has been converted to lower case, and the non-alphabetic letters has been removed before the mono-alphabetic cipher.

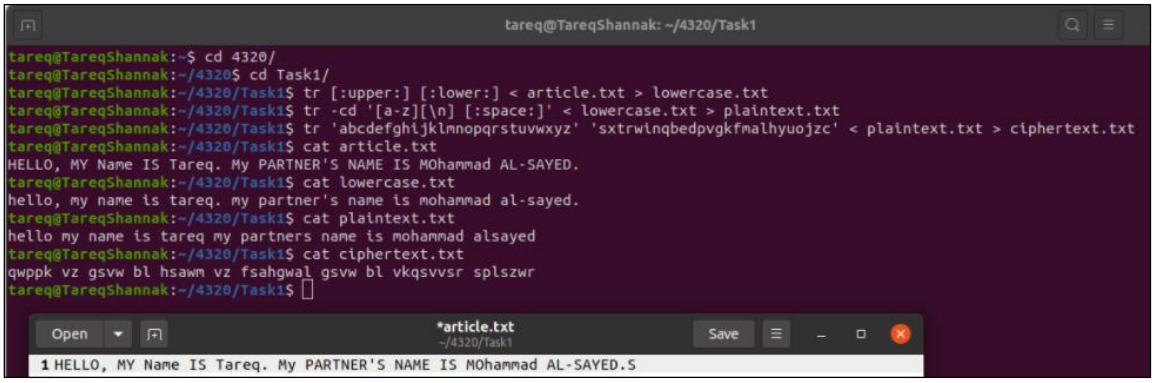


Figure 2 - Converting to lowercase and removing non-alphabetic letters

Step 3

In this task, the cipher.txt was converted to plain.txt using the frequency analysis of letters, in order to retrieve the original message that was sent, we used Python Code to do the conversion of letters (You can find it in the appendix). First, the cipher 'ytn' was used as it represents 'THE' from English, so every Y in the cipher is a T, every T is an H, and every N is an E. Another way, we counted how many times each letter was repeated, and it was 'n' in the

cipher, and the most repetitive letter in English is an ‘E’ so it means we are on the right path to decryption. So far, we’ve the solution for 3 letters in the cipher, and when we changed them then looked for another word, we found the following:

- The letter ‘v’ in the cipher was found alone, so what any other letter in English than ‘A’ can be alone. Thus, every ‘v’ in the cipher was changed to ‘A’, then we complied again.
- The word ‘Tx’ (Capital letters represent converted English letters) was found, what could it be other than ‘TO’.
- The words ‘uO’ and ‘Ou’ were found in the cipher, so what we predicted that ‘u’ is an ‘N’ which then we will have ‘NO’ and ‘ON’, so every ‘u’ is an ‘N’.
- The word ‘ANp’ was found, what could it be other than ‘AND’, so every ‘p’ is a ‘D’.
- The sentence ‘TO gE A’ was found, since every letter is converted except for ‘g’ and considering the sentence meaning, we predicted it to be ‘TO BE A’, so every ‘g’ is a ‘B’.
- The word ‘HOl’ was found, what could it be other than ‘HOW’, so every ‘l’ is a ‘W’.
- The word ‘OTHEh’ was found, what could it be other than ‘OTHER’, so every ‘h’ is an ‘R’.
- The word ‘Aii’ was found, what could it be other than ‘ALL’, so every ‘I’ is an ‘L’.
- The word ‘mT’ was found, and since ‘A’ is already found, it couldn’t be ‘AT’, so what can it be other than ‘IT’, so every ‘m’ is an ‘I’.
- The word ‘WAq’ was found, it couldn’t be ‘WAR’ since we found the replacement for ‘R, so what could it be other than ‘WAS’, so every ‘q’ is an ‘S’.
- The word ‘RIrHT’ was found, what could it be other than ‘RIGHT’, so every ‘r’ is a ‘G’.
- The word ‘ABOzT’ was found, what could it be other than ‘ABOUT’, so every ‘z’ is a ‘U’.
- The word ‘AbTER’ was found, what could it be other than ‘AFTER’, so every ‘b is an ‘F’.

- The word ‘SUNDAd’ was found, what could it be other than ‘SUNDAY’, so every ‘d’ is a ‘Y’.
- The word ‘THANsS’ was found, what could it be other than ‘THANKS’, so every ‘s’ is a ‘K’.
- The word ‘DREAc’ was found, what could it be other than ‘DREAM’, so every ‘c’ is an ‘M’.
- The word ‘RAaE’ was found, what could it be other than ‘RACE’, so every ‘a’ is a ‘C’.
- The word ‘TRIe’ was found, what could it be other than ‘TRIP’, so every ‘e’ is a ‘P’.
- The word ‘AfOID’ was found, what could it be other than ‘AVOID’, so every ‘f’ is a ‘V’.
- The word ‘oUST’ was found, what could it be other than ‘JUST’, so every ‘o’ is a ‘J’.
- The word ‘EkSTRA’ was found, what could it be other than ‘EXTRA’, so every ‘k is an ‘X’.
- The word ‘EjUALLY’ was found, what could it be other than ‘EQUALLY’, so every ‘j is a ‘Q’.

Thus, the key found is as the following = “VGAPNBRTMOSICUXEJHQYZFLKDW”

You can find below the plain.txt after the decryption (You can find it as text in the appendix):

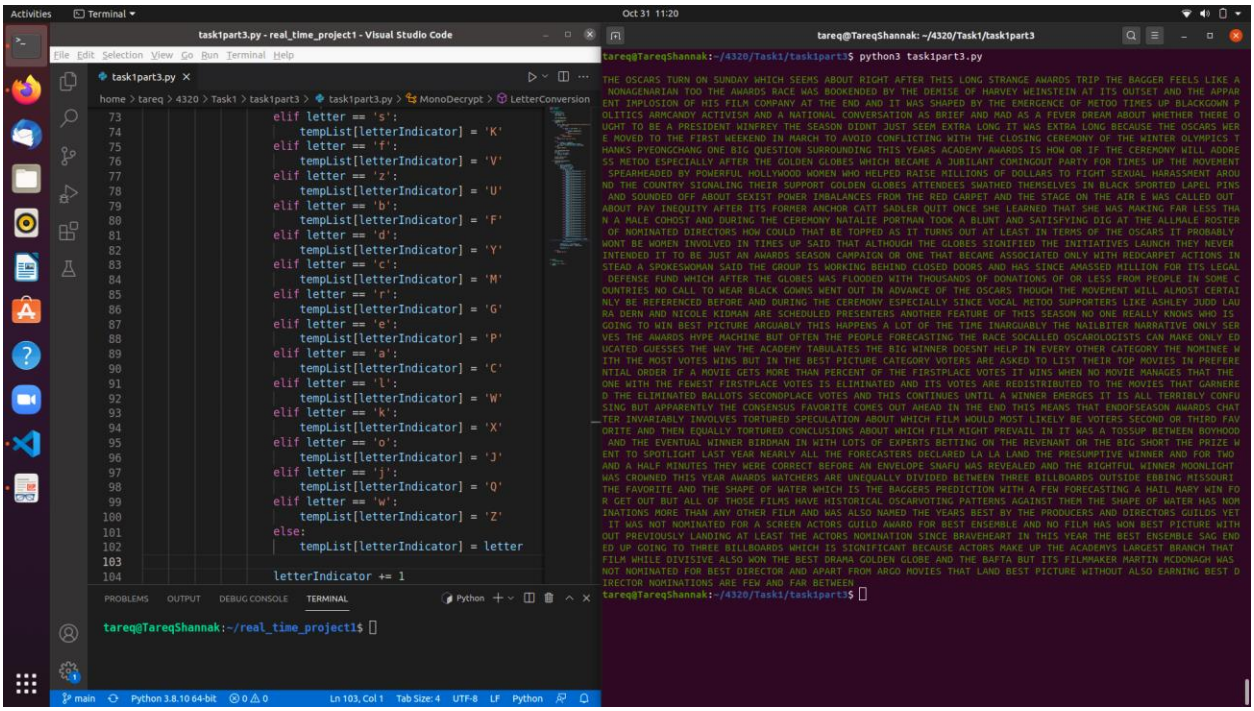


Figure 3 - The Decrypted Message

Task 2: Encryption using Different Ciphers and Modes

In this task, we tried three different block ciphers: AES 128 bits Cipher Block Chain, Blow Fish Cipher Block Chain and Cipher Feedback. We can see the difference between the ciphers in the cipher texts in the figure below.

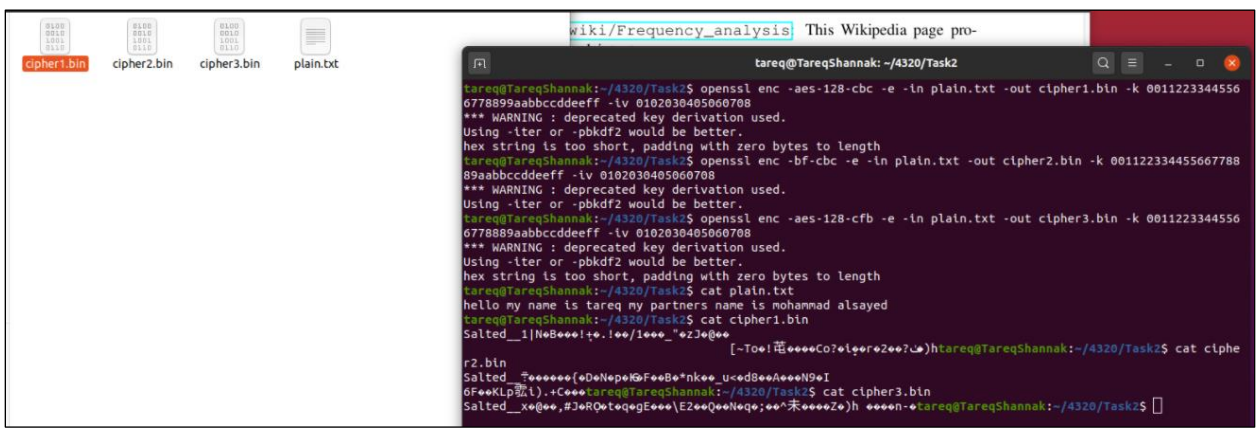


Figure 4 - Different Cipher Modes

Task 3: Encryption Mode – ECB vs. CBC

We encrypted the penguin image by 2 ciphers: Electronic Codebook Mode and Cipher Block Chaining Mode with Initialization Vector (IV). We encrypted the body of the image and put the original header.

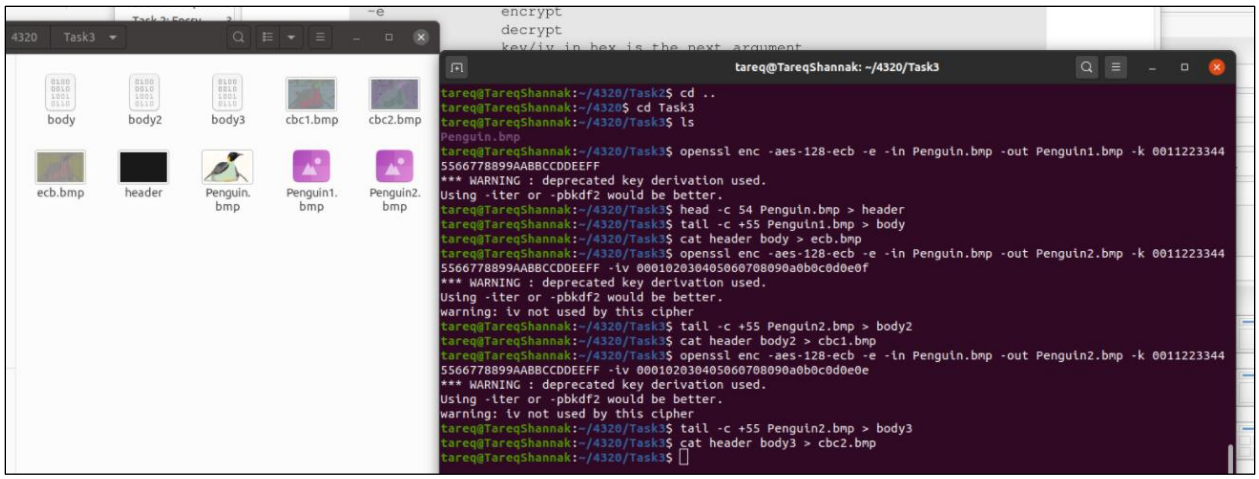


Figure 5 - Encrypting IMG

The figure below shows the encrypted image using ECB.

Figure 6 - Encrypted IMG using ECB

The figure below shows the encryption using CBC with certain IV, and the next figure shows the encryption with another IV. We can conclude that the encryption differs for the same message in different IVs which is good.

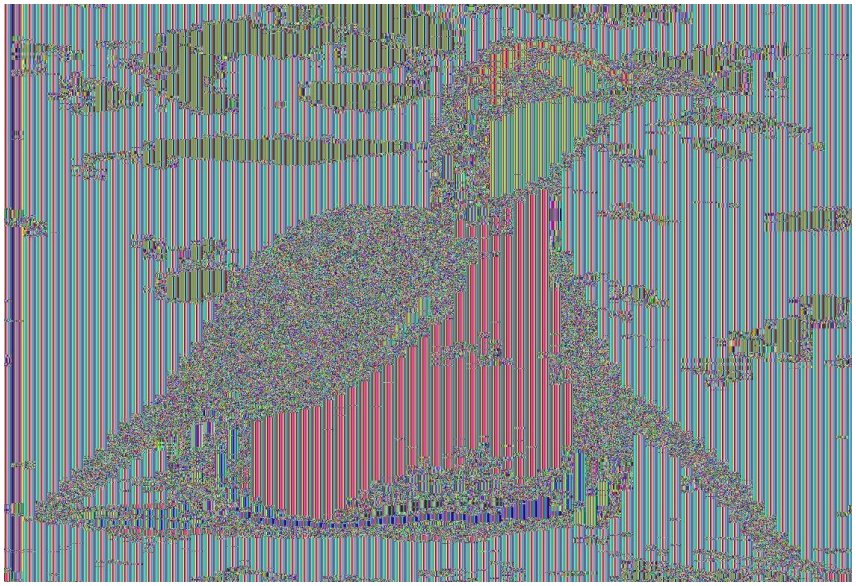


Figure 7 - Encrypted IMG with IV using CBC

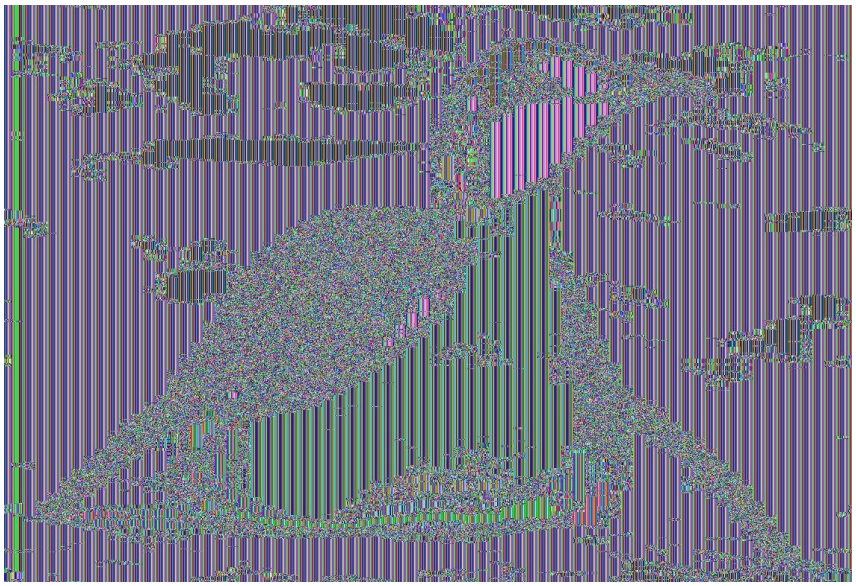


Figure 8 - Encrypted IMG with different IV using CBC

Task 4: Padding

In this task we encrypted three files with different lengths using three ciphers: ECB, CBC and CFB.

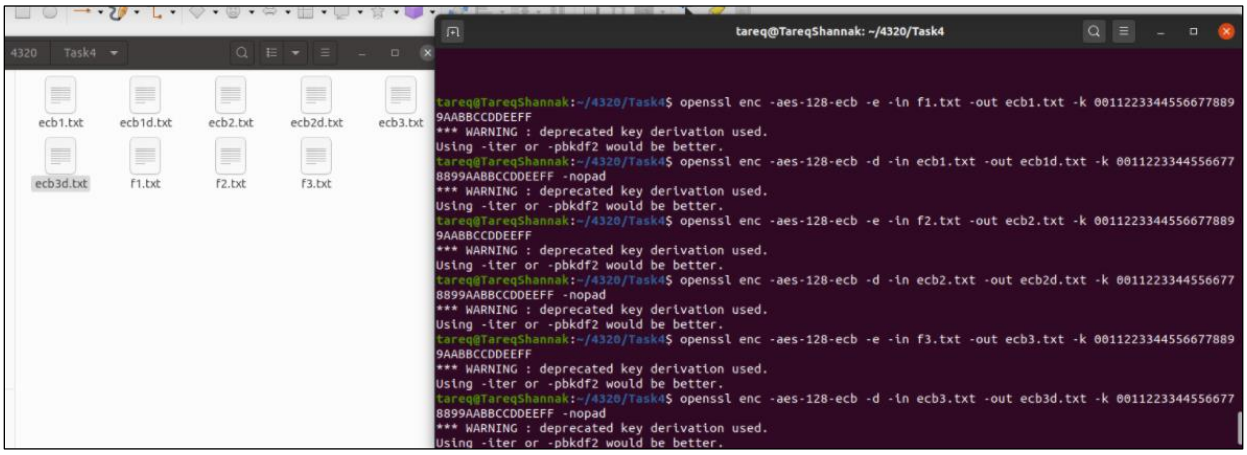


Figure 9 - Encrypting file using ECB

In CBC, the first encrypted file added 11 bytes of 0X0B as padding data because there is 11 bytes (B in HEX=11 in decimal) to complete block size of 16 bytes. The second file add 6 bytes of 0X06 as padding data. The third file's length is multiple of the block size 16, so the padding bytes are 16 bytes of 0X10 (10 in HEX=16 in decimal) as shown in the figure below.

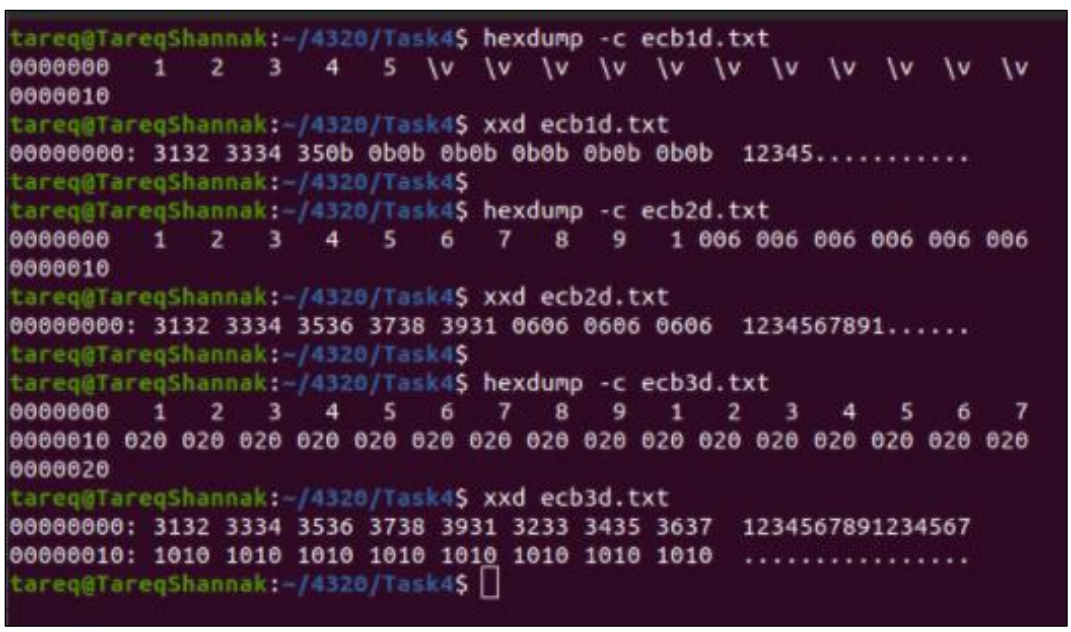


Figure 10 - Padding files using CBC

The CBC is the same idea in padding of the EBC as shown in the figure below.

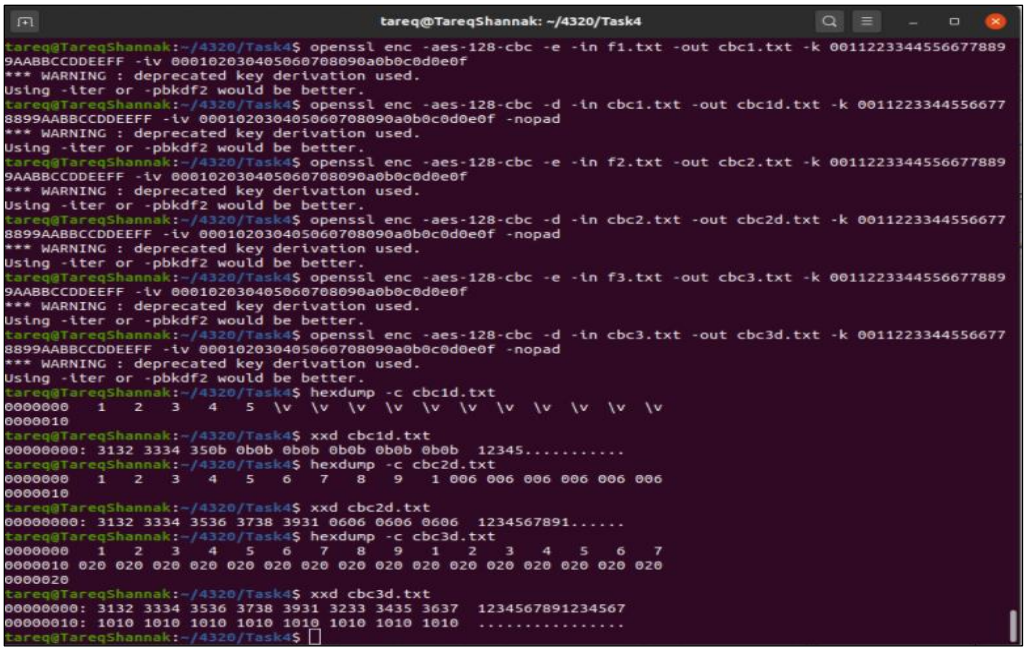


Figure 11 - Encrypting File with CBC

In CFB, the cipher text size is same as plain text size, so it does not need padding unlike CBC and ECB as shown in the figure below.

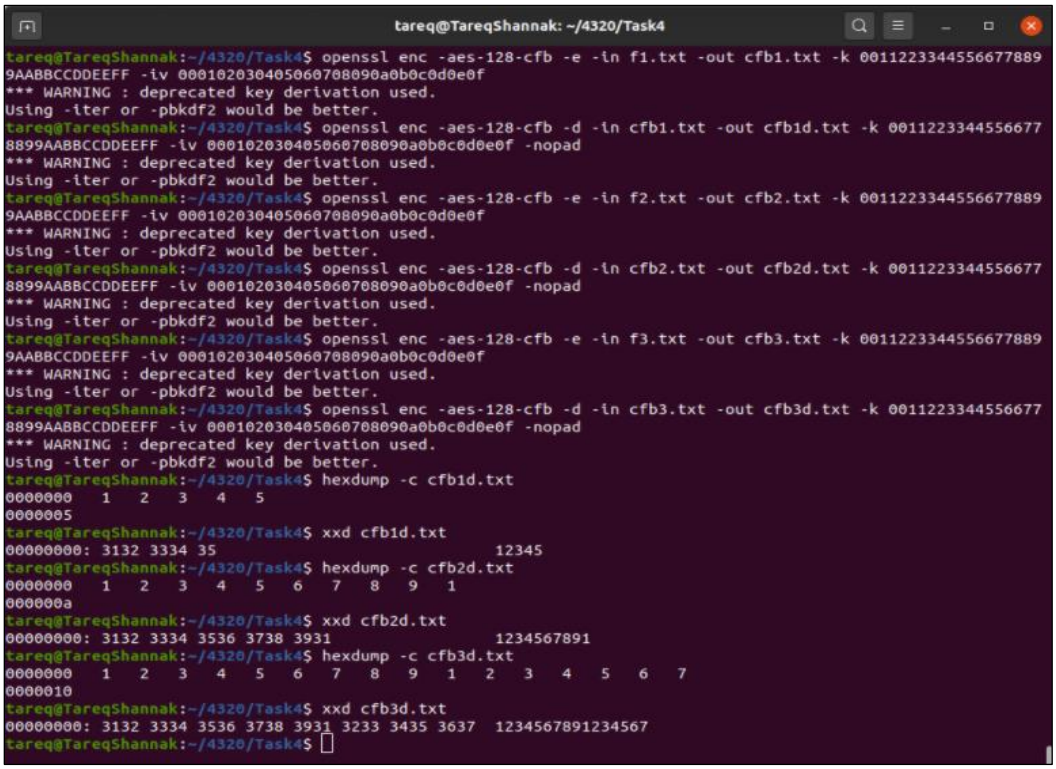
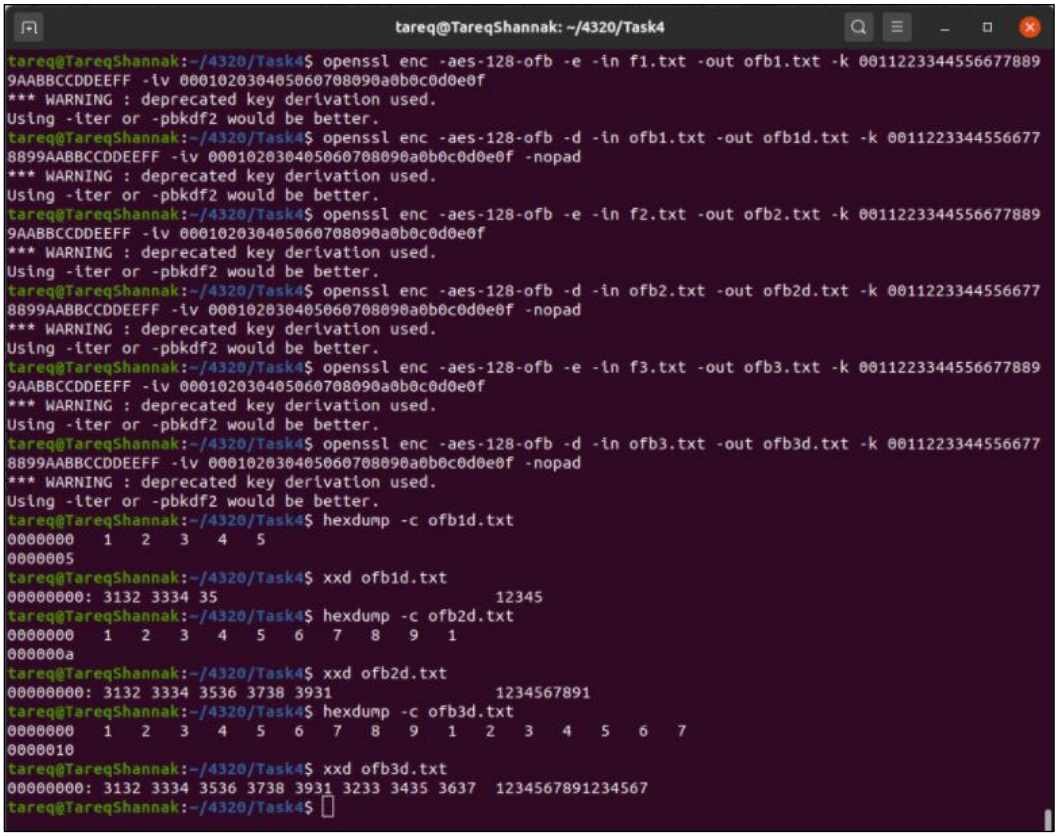


Figure 12 - Encrypting File with CFB

OFB is the same idea in padding as CFB since it does not padding as shown in the figure below.

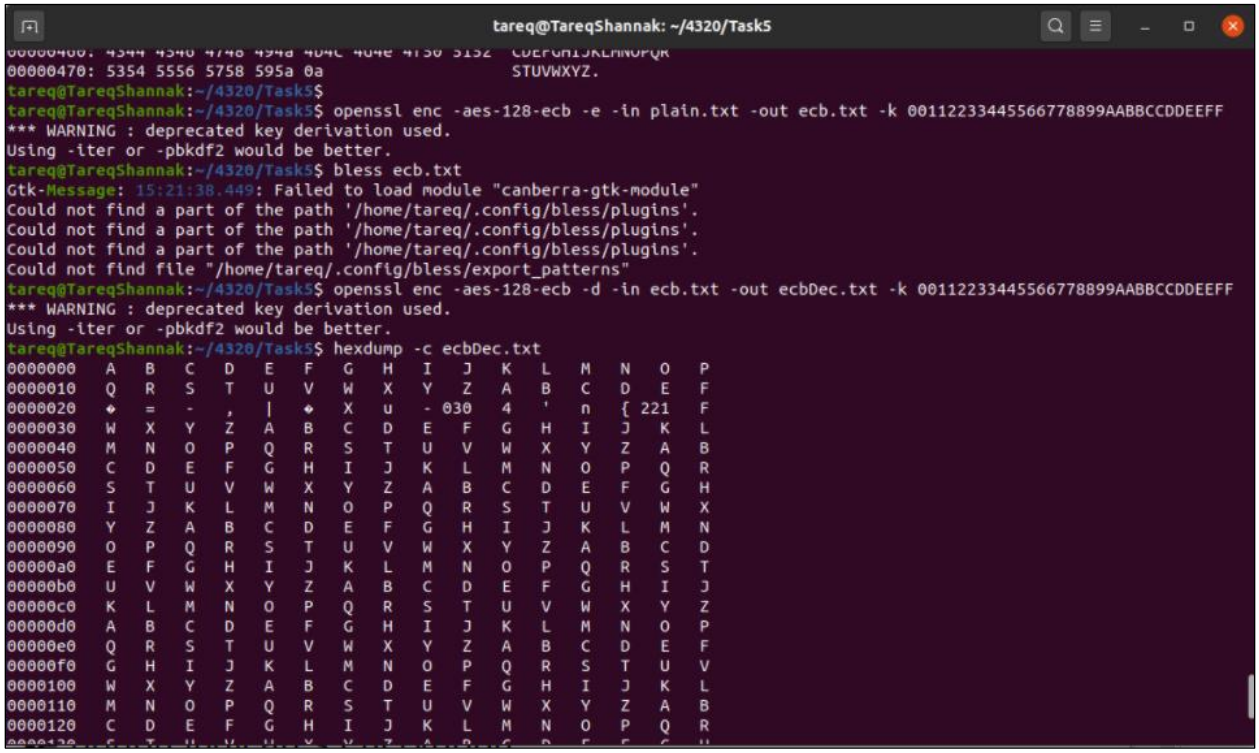


```
tareq@TareqShannak: ~/4320/Task4
tareq@TareqShannak:~/4320/Task4$ openssl enc -aes-128-ofb -e -in f1.txt -out ofb1.txt -k 0011223344556677889
9AABBCCDDEEFF -iv 000102030405060708090a0b0c0d0e0f
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task4$ openssl enc -aes-128-ofb -d -in ofb1.txt -out ofb1d.txt -k 0011223344556677
8899AABBCCDDEEFF -iv 000102030405060708090a0b0c0d0e0f -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task4$ openssl enc -aes-128-ofb -e -in f2.txt -out ofb2.txt -k 0011223344556677889
9AABBCCDDEEFF -iv 000102030405060708090a0b0c0d0e0f
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task4$ openssl enc -aes-128-ofb -d -in ofb2.txt -out ofb2d.txt -k 0011223344556677
8899AABBCCDDEEFF -iv 000102030405060708090a0b0c0d0e0f -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task4$ openssl enc -aes-128-ofb -e -in f3.txt -out ofb3.txt -k 0011223344556677889
9AABBCCDDEEFF -iv 000102030405060708090a0b0c0d0e0f
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task4$ openssl enc -aes-128-ofb -d -in ofb3.txt -out ofb3d.txt -k 0011223344556677
8899AABBCCDDEEFF -iv 000102030405060708090a0b0c0d0e0f -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task4$ hexdump -c ofb1d.txt
0000000   1   2   3   4   5
0000005
tareq@TareqShannak:~/4320/Task4$ xxd ofb1d.txt
00000000: 3132 3334 35                             12345
tareq@TareqShannak:~/4320/Task4$ hexdump -c ofb2d.txt
0000000   1   2   3   4   5   6   7   8   9   1
000000a
tareq@TareqShannak:~/4320/Task4$ xxd ofb2d.txt
00000000: 3132 3334 3536 3738 3931                 1234567891
tareq@TareqShannak:~/4320/Task4$ hexdump -c ofb3d.txt
0000000   1   2   3   4   5   6   7   8   9   1   2   3   4   5   6   7
0000010
tareq@TareqShannak:~/4320/Task4$ xxd ofb3d.txt
00000000: 3132 3334 3536 3738 3931 3233 3435 3637  1234567891234567
tareq@TareqShannak:~/4320/Task4$
```

Figure 13 - OFB Padding

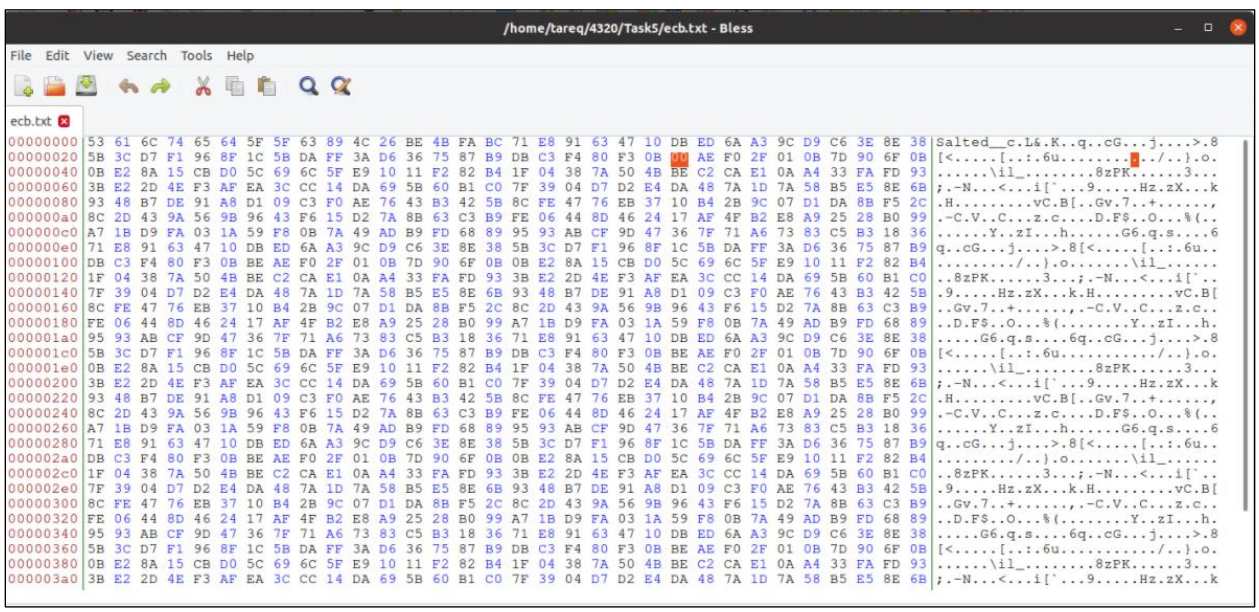
Task 5: Error Propagation – Corrupted Cipher Text

In ECB, the block size completely has been corrupted.



```
tareq@TareqShannak: ~/4320/Task5
00000470: 5354 5556 5758 595a 0a                    STUVWXYZ.
tareq@TareqShannak:~/4320/Task5$
tareq@TareqShannak:~/4320/Task5$ openssl enc -aes-128-ecb -e -in plain.txt -out ecb.txt -k 00112233445566778899AABBCCDDEEFF
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task5$ bless ecb.txt
Gtk-Message: 15:21:38.449: Failed to load module "canberra-gtk-module"
Could not find a part of the path '/home/tareq/.config/bless/plugins'.
Could not find a part of the path '/home/tareq/.config/bless/plugins'.
Could not find a part of the path '/home/tareq/.config/bless/plugins'.
Could not find file "/home/tareq/.config/bless/export_patterns"
tareq@TareqShannak:~/4320/Task5$ openssl enc -aes-128-ecb -d -in ecb.txt -out ecbDec.txt -k 00112233445566778899AABBCCDDEEFF
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task5$ hexdump -c ecbDec.txt
0000000   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P
0000010   Q   R   S   T   U   V   W   X   Y   Z   A   B   C   D   E   F
0000020   ◆   =   -   ,   |   ◆   X   u   - 030   4   '   n   { 221   F
0000030   W   X   Y   Z   A   B   C   D   E   F   G   H   I   J   K   L
0000040   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z   A   B
0000050   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R
0000060   S   T   U   V   W   X   Y   Z   A   B   C   D   E   F   G   H
0000070   I   J   K   L   M   N   O   P   Q   R   S   T   U   V   W   X
0000080   Y   Z   A   B   C   D   E   F   G   H   I   J   K   L   M   N
0000090   O   P   Q   R   S   T   U   V   W   X   Y   Z   A   B   C   D
00000a0   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
00000b0   U   V   W   X   Y   Z   A   B   C   D   E   F   G   H   I   J
00000c0   K   L   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z
00000d0   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P
00000e0   Q   R   S   T   U   V   W   X   Y   Z   A   B   C   D   E   F
00000f0   G   H   I   J   K   L   M   N   O   P   Q   R   S   T   U   V
0000100   W   X   Y   Z   A   B   C   D   E   F   G   H   I   J   K   L
0000110   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z   A   B
0000120   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R
```

Figure 14 – Decryption the corrupted cipher text in ECB



```
/home/tareq/4320/Task5/ecb.txt - Bless
File  Edit  View  Search  Tools  Help
ecb.txt
00000000 53 61 6C 74 65 64 5F 5F 63 89 4C 26 BE 4B FA BC 71 E8 91 63 47 10 DB ED 6A A3 9C D9 C6 3E 8E 38 |Salted__c.L&.K..q..cG...j....>.8
00000020 5B 3C D7 F1 96 8F 1C 5B DA FF 3A D6 36 75 87 B9 DB C3 F4 80 F3 0B 00 AE F0 2F 01 0B 7D 90 6F 0B |[<.....[..:.6u.........../..}.o.
00000040 0B E2 8A 15 CB D0 5C 69 6C 5F E9 10 11 F2 82 B4 1F 04 38 7A 50 4B BE C2 CA E1 0A A4 33 FA FD 93 |......\il_........8zPK......3...
00000060 3B E2 2D 4E F3 AF EA 3C CC 14 DA 69 5B 60 B1 C0 7F 39 04 D7 D2 E4 DA 48 7A 1D 7A 58 B5 E5 8E 6B |;.-N...<...i[`...9.....Hz.zX...k
00000080 93 48 B7 DE 91 A8 D1 09 C3 F0 AE 76 43 B3 42 5B 8C FE 47 76 EB 37 10 B4 2B 9C 07 D1 DA 8B F5 2C |.H.........vC.B[..Gv.7..+......,
000000a0 8C 2D 43 9A 56 9B 96 43 F6 15 D2 7A 8B 63 C3 B9 FE 06 44 8D 46 24 17 AF 4F B2 E8 A9 25 28 B0 99 |.-C.V..C...z.c....D.F$..O...%(..
000000c0 A7 1B D9 FA 03 1A 59 F8 0B 7A 49 AD B9 FD 68 89 95 93 AB CF 9D 47 36 7F 71 A6 73 83 C5 B3 18 36 |......Y..zI...h......G6.q.s....6
000000e0 71 E8 91 63 47 10 DB ED 6A A3 9C D9 C6 3E 8E 38 5B 3C D7 F1 96 8F 1C 5B DA FF 3A D6 36 75 87 B9 |q..cG...j....>.8[<.....[..:.6u..
00000100 DB C3 F4 80 F3 0B BE AE F0 2F 01 0B 7D 90 6F 0B 0B E2 8A 15 CB D0 5C 69 6C 5F E9 10 11 F2 82 B4 |...../..}.o.......\il_......
00000120 1F 04 38 7A 50 4B BE C2 CA E1 0A A4 33 FA FD 93 3B E2 2D 4E F3 AF EA 3C CC 14 DA 69 5B 60 B1 C0 |..8zPK......3...;.-N...<...i[`..
00000140 7F 39 04 D7 D2 E4 DA 48 7A 1D 7A 58 B5 E5 8E 6B 93 48 B7 DE 91 A8 D1 09 C3 F0 AE 76 43 B3 42 5B |.9.....Hz.zX...k.H.........vC.B[
00000160 8C FE 47 76 EB 37 10 B4 2B 9C 07 D1 DA 8B F5 2C 8C 2D 43 9A 56 9B 96 43 F6 15 D2 7A 8B 63 C3 B9 |..Gv.7..+......,.-C.V..C...z.c..
00000180 FE 06 44 8D 46 24 17 AF 4F B2 E8 A9 25 28 B0 99 A7 1B D9 FA 03 1A 59 F8 0B 7A 49 AD B9 FD 68 89 |..D.F$..O...%(........Y..zI...h.
000001a0 95 93 AB CF 9D 47 36 7F 71 A6 73 83 C5 B3 18 36 71 E8 91 63 47 10 DB ED 6A A3 9C D9 C6 3E 8E 38 |.....G6.q.s....6q..cG...j....>.8
000001c0 5B 3C D7 F1 96 8F 1C 5B DA FF 3A D6 36 75 87 B9 DB C3 F4 80 F3 0B BE AE F0 2F 01 0B 7D 90 6F 0B |[<.....[..:.6u.........../..}.o.
000001e0 0B E2 8A 15 CB D0 5C 69 6C 5F E9 10 11 F2 82 B4 1F 04 38 7A 50 4B BE C2 CA E1 0A A4 33 FA FD 93 |......\il_........8zPK......3...
00000200 3B E2 2D 4E F3 AF EA 3C CC 14 DA 69 5B 60 B1 C0 7F 39 04 D7 D2 E4 DA 48 7A 1D 7A 58 B5 E5 8E 6B |;.-N...<...i[`...9.....Hz.zX...k
00000220 93 48 B7 DE 91 A8 D1 09 C3 F0 AE 76 43 B3 42 5B 8C FE 47 76 EB 37 10 B4 2B 9C 07 D1 DA 8B F5 2C |.H.........vC.B[..Gv.7..+......,
00000240 8C 2D 43 9A 56 9B 96 43 F6 15 D2 7A 8B 63 C3 B9 FE 06 44 8D 46 24 17 AF 4F B2 E8 A9 25 28 B0 99 |.-C.V..C...z.c....D.F$..O...%(..
00000260 A7 1B D9 FA 03 1A 59 F8 0B 7A 49 AD B9 FD 68 89 95 93 AB CF 9D 47 36 7F 71 A6 73 83 C5 B3 18 36 |......Y..zI...h......G6.q.s....6
00000280 71 E8 91 63 47 10 DB ED 6A A3 9C D9 C6 3E 8E 38 5B 3C D7 F1 96 8F 1C 5B DA FF 3A D6 36 75 87 B9 |q..cG...j....>.8[<.....[..:.6u..
000002a0 DB C3 F4 80 F3 0B BE AE F0 2F 01 0B 7D 90 6F 0B 0B E2 8A 15 CB D0 5C 69 6C 5F E9 10 11 F2 82 B4 |...../..}.o.......\il_......
000002c0 1F 04 38 7A 50 4B BE C2 CA E1 0A A4 33 FA FD 93 3B E2 2D 4E F3 AF EA 3C CC 14 DA 69 5B 60 B1 C0 |..8zPK......3...;.-N...<...i[`..
000002e0 7F 39 04 D7 D2 E4 DA 48 7A 1D 7A 58 B5 E5 8E 6B 93 48 B7 DE 91 A8 D1 09 C3 F0 AE 76 43 B3 42 5B |.9.....Hz.zX...k.H.........vC.B[
00000300 8C FE 47 76 EB 37 10 B4 2B 9C 07 D1 DA 8B F5 2C 8C 2D 43 9A 56 9B 96 43 F6 15 D2 7A 8B 63 C3 B9 |..Gv.7..+......,.-C.V..C...z.c..
00000320 FE 06 44 8D 46 24 17 AF 4F B2 E8 A9 25 28 B0 99 A7 1B D9 FA 03 1A 59 F8 0B 7A 49 AD B9 FD 68 89 |..D.F$..O...%(........Y..zI...h.
00000340 95 93 AB CF 9D 47 36 7F 71 A6 73 83 C5 B3 18 36 71 E8 91 63 47 10 DB ED 6A A3 9C D9 C6 3E 8E 38 |.....G6.q.s....6q..cG...j....>.8
00000360 5B 3C D7 F1 96 8F 1C 5B DA FF 3A D6 36 75 87 B9 DB C3 F4 80 F3 0B BE AE F0 2F 01 0B 7D 90 6F 0B |[<.....[..:.6u.........../..}.o.
00000380 0B E2 8A 15 CB D0 5C 69 6C 5F E9 10 11 F2 82 B4 1F 04 38 7A 50 4B BE C2 CA E1 0A A4 33 FA FD 93 |......\il_........8zPK......3...
000003a0 3B E2 2D 4E F3 AF EA 3C CC 14 DA 69 5B 60 B1 C0 7F 39 04 D7 D2 E4 DA 48 7A 1D 7A 58 B5 E5 8E 6B |;.-N...<...i[`...9.....Hz.zX...k
```

Figure 15 – Change the 55th byte

Also in CBC and CFB, the block size (16 Bytes) has been corrupted.

```
tareq@TareqShannak: ~/4320/Task5

tareq@TareqShannak:~/4320/Task5$ openssl enc -aes-128-cbc -e -in plain.txt -out cbc.txt -k 00112233445566778899AABBCCDDEEFF -
iv 000102030405060708090a0b0c0d0e0f
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task5$ bless cbc.txt
Gtk-Message: 15:35:06.532: Failed to load module "canberra-gtk-module"
Could not find a part of the path '/home/tareq/.config/bless/plugins'.
Could not find a part of the path '/home/tareq/.config/bless/plugins'.
Could not find a part of the path '/home/tareq/.config/bless/plugins'.
Could not find file "/home/tareq/.config/bless/export_patterns"
tareq@TareqShannak:~/4320/Task5$ openssl enc -aes-128-cbc -d -in cbc.txt -out cbcDec.txt -k 00112233445566778899AABBCCDDEEFF
-iv 000102030405060708090a0b0c0d0e0f
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task5$ hexdump -c cbcDec.txt
0000000   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P
0000010   Q   R   S   T   U   V   W   X   Y   Z   A   B   C   D   E   F
0000020   ◆   ◆   y   ◆  \n   T  \t 227   2   ?   i 210   ◆   f  \0   ◆
0000030   W   X   Y   Z   A   B   4   D   E   F   G   H   I   J   K   L
0000040   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z   A   B
0000050   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R
0000060   S   T   U   V   W   X   Y   Z   A   B   C   D   E   F   G   H
0000070   I   J   K   L   M   N   O   P   Q   R   S   T   U   V   W   X
0000080   Y   Z   A   B   C   D   E   F   G   H   I   J   K   L   M   N
0000090   O   P   Q   R   S   T   U   V   W   X   Y   Z   A   B   C   D
00000a0   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
00000b0   U   V   W   X   Y   Z   A   B   C   D   E   F   G   H   I   J
00000c0   K   L   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z
00000d0   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P
00000e0   Q   R   S   T   U   V   W   X   Y   Z   A   B   C   D   E   F
00000f0   G   H   I   J   K   L   M   N   O   P   Q   R   S   T   U   V
0000100   W   X   Y   Z   A   B   C   D   E   F   G   H   I   J   K   L
0000110   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z   A   B
0000120   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R
```

Figure 16 - Decryption the corrupted cipher text in CBC

```
tareq@TareqShannak: ~/4320/Task5

tareq@TareqShannak:~/4320/Task5$
tareq@TareqShannak:~/4320/Task5$ openssl enc -aes-128-cfb -e -in plain.txt -out cfb.txt -k 00112233445566778899AABBCCDDEEFF -
iv 000102030405060708090a0b0c0d0e0f
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task5$ bless cfb.txt
Gtk-Message: 15:36:58.255: Failed to load module "canberra-gtk-module"
Could not find a part of the path '/home/tareq/.config/bless/plugins'.
Could not find a part of the path '/home/tareq/.config/bless/plugins'.
Could not find a part of the path '/home/tareq/.config/bless/plugins'.
Could not find file "/home/tareq/.config/bless/export_patterns"
tareq@TareqShannak:~/4320/Task5$ openssl enc -aes-128-cfb -d -in cfb.txt -out cfbDec.txt -k 00112233445566778899AABBCCDDEEFF
-iv 000102030405060708090a0b0c0d0e0f
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tareq@TareqShannak:~/4320/Task5$ hexdump -c cfbDec.txt
0000000   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P
0000010   Q   R   S   T   U   V   W   X   Y   Z   A   B   C   D   E   F
0000020   G   H   I   J   K   L 177   N   O   P   Q   R   S   T   U   V
0000030   ◆   m   ; 223   ◆   ◆   >   ◆   ◆   .   w   ◆   2   ◆ 202   u
0000040   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z   A   B
0000050   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R
0000060   S   T   U   V   W   X   Y   Z   A   B   C   D   E   F   G   H
0000070   I   J   K   L   M   N   O   P   Q   R   S   T   U   V   W   X
0000080   Y   Z   A   B   C   D   E   F   G   H   I   J   K   L   M   N
0000090   O   P   Q   R   S   T   U   V   W   X   Y   Z   A   B   C   D
00000a0   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T
00000b0   U   V   W   X   Y   Z   A   B   C   D   E   F   G   H   I   J
00000c0   K   L   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z
00000d0   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P
00000e0   Q   R   S   T   U   V   W   X   Y   Z   A   B   C   D   E   F
00000f0   G   H   I   J   K   L   M   N   O   P   Q   R   S   T   U   V
0000100   W   X   Y   Z   A   B   C   D   E   F   G   H   I   J   K   L
0000110   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z   A   B
0000120   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R
```

Figure 17 - Decryption the corrupted cipher text in CFB

In OFB, the effect on the corrupted cipher text is only the corrupted byte without changing anything else.

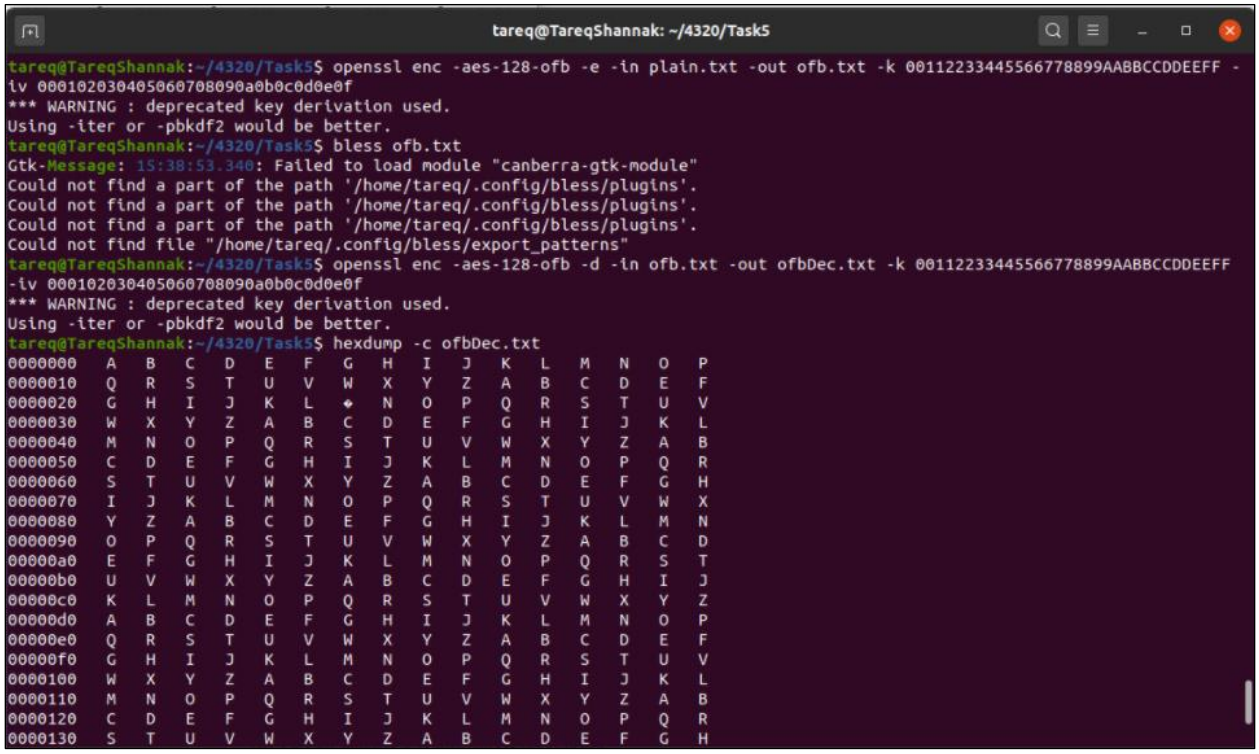


Figure 18 - Decryption the corrupted cipher text in OFB

Task 6: Initial Vector (IV) and Common Mistakes

Part 1

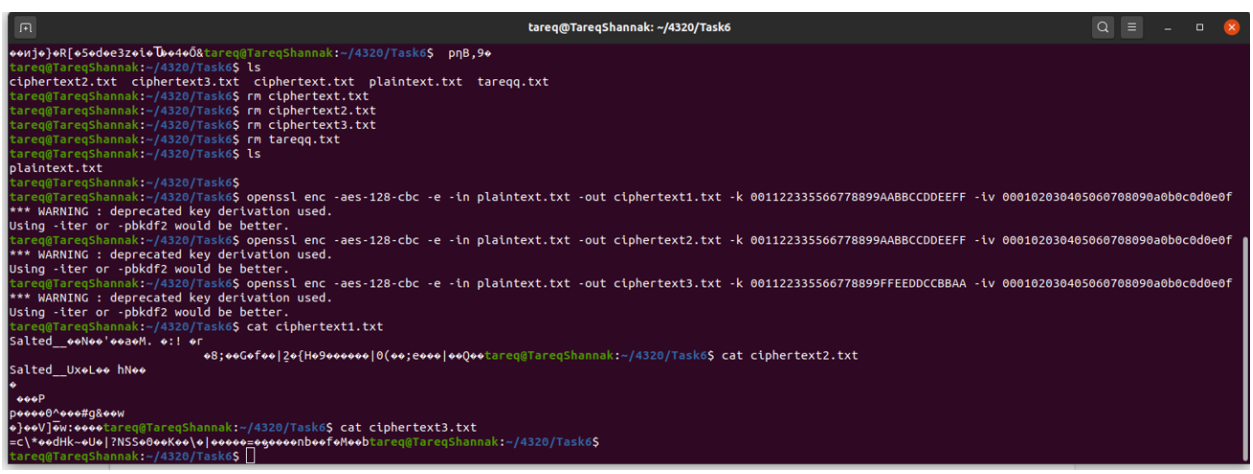


Figure 19 - Different and Same IVs

Part 2

```
home > tareq > 4320 > Task6 > task6part2 > sample_code.py > ...
#!/usr/bin/python3

# XOR two bytearrays
def xor(first, second):
    return bytearray(x^y for x,y in zip(first, second))

MSG   = "A message"
HEX_1 = "aabbccddeeff1122334455"
HEX_2 = "1122334455778800aabbdd"

# Convert ascii string to bytearray
D1 = bytes(MSG, 'utf-8')

# Convert hex string to bytearray
D2 = bytearray.fromhex(HEX_1)
D3 = bytearray.fromhex(HEX_2)


r2 = xor(D2, D3)
r3 = xor(r2, D1)

print(D1.hex())
print(r2.hex())
print(r3.hex())
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

tareq@TareqShannak:~$ /bin/python3 /home/tareq/4320/Task6/task6part2/sample_code.py
41206d657373616765
bb99ff99bb88992299ff88
fab992fcc8fbf845fc
tareq@TareqShannak:~$ S
```

Figure 20 - Obtain P2 from P1, C1 and C2

Part 3

After padding, it seems that the word is YES which we obtained after predicting the next IV.

```
File  Actions  Edit  View  Help
┌──(user㉿kali)-[~]
└─$ cd Downloads/
┌──(user㉿kali)-[~/Downloads]
└─$ cd Labsetup/
┌──(user㉿kali)-[~/Downloads/Labsetup]
└─$ nc 10.9.0.80 3000
Bob's secret message is either "Yes" or "No", without quotations.
Bob's ciphertex: 7c08b24ba57bc01d65e56f4909f676d1
The IV used   : e1f30f7153ae8e948d4269ad0be83594

Next IV       : a7c1efe053ae8e948d4269ad0be83594
Your plaintext : 1123344aabbccdd
Invalid hex string

Next IV       : 0a6c272c54ae8e948d4269ad0be83594
Your plaintext : 11223344aabbccdd
Your ciphertext: 7de003674ff4c862312538cf65d465d2

Next IV       : fc5b4caa54ae8e948d4269ad0be83594
Your plaintext :
```

Figure 21 - Knowing the word

Appendix

Python Code:

```
import operator
import string

class MonoDecrypt:
        def __init__(self):
                self.Data = []
                self.Count = {}

                self.FileReading()
                #print(self.Count)

        def FileReading(self):
                TempData = []
                try:
                        File = open('Labsetup/Files/ciphertext.txt', '+r')
                        for line in File:
                                try:
                                        for Word in line.split(' '):
                                                try:
                                                        TempData.append(Word)
                                                except:
                                                        print("Unsuccessful")
                                except:
                                        print("ERROR")
                except:
                        print("Error with the file")

                try:
                        for element in TempData:
                                self.Data.append(element.strip())
                        self.Data = list(filter(None, self.Data))
                except:
                        print("ERROR")

                self.WordsCount()

        def WordsCount(self):
                tempData = {}
                try:
                        for word in self.Data:
                                for letter in word:
                                        if letter in tempData.keys():
                                                tempData[letter] += 1
                                        else:
                                                tempData[letter] = 1
                        self.Count = dict(sorted(tempData.items(), key=lambda x:x[1],
reverse=True))


                except:
                        print("ERROR")

                self.LetterConversion()

        def LetterConversion(self):
                Indicator = 0
                try:
                        for word in self.Data:
                                letterIndicator = 0
```

```
tempList = list(word)
for letter in word:
        if letter == 'n':
                tempList[letterIndicator] = 'E'
        elif letter == 't':
                tempList[letterIndicator] = 'H'
        elif letter == 'y':
                tempList[letterIndicator] = 'T'
        elif letter == 'v':
                tempList[letterIndicator] = 'A'
        elif letter == 'x':
                tempList[letterIndicator] = 'O'
        elif letter == 'u':
                tempList[letterIndicator] = 'N'
        elif letter == 'b':
                tempList[letterIndicator] = 'F'
        elif letter == 'p':
                tempList[letterIndicator] = 'D'
        elif letter == 'g':
                tempList[letterIndicator] = 'B'
        elif letter == 'h':
                tempList[letterIndicator] = 'R'
        elif letter == 'm':
                tempList[letterIndicator] = 'I'
        elif letter == 'i':
                tempList[letterIndicator] = 'L'
        elif letter == 'q':
                tempList[letterIndicator] = 'S'
        elif letter == 's':
                tempList[letterIndicator] = 'K'
        elif letter == 'f':
                tempList[letterIndicator] = 'V'
        elif letter == 'z':
                tempList[letterIndicator] = 'U'
        elif letter == 'd':
                tempList[letterIndicator] = 'Y'
        elif letter == 'c':
                tempList[letterIndicator] = 'M'
        elif letter == 'r':
                tempList[letterIndicator] = 'G'
        elif letter == 'e':
                tempList[letterIndicator] = 'P'
        elif letter == 'a':
                tempList[letterIndicator] = 'C'
        elif letter == 'l':
                tempList[letterIndicator] = 'W'
        elif letter == 'k':
                tempList[letterIndicator] = 'X'
        elif letter == 'o':
                tempList[letterIndicator] = 'J'
        elif letter == 'j':
                tempList[letterIndicator] = 'Q'
        elif letter == 'w':
                tempList[letterIndicator] = 'Z'
        else:
                tempList[letterIndicator] = letter

        letterIndicator += 1

tempString = "".join(tempList)
self.Data[Indicator] = tempString

Indicator += 1
```

```
                        print(self.Data)

                except:
                        print("Error")

def main():
        C = MonoDecrypt()

if __name__ == "__main__":
        main()
```

Recovered 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