

Lab 4

Crypto Lab - Secret-Key Encryption

CPSC 353-01
Introduction to Computer Security
Professor Kenytt Avery

By Danh Pham

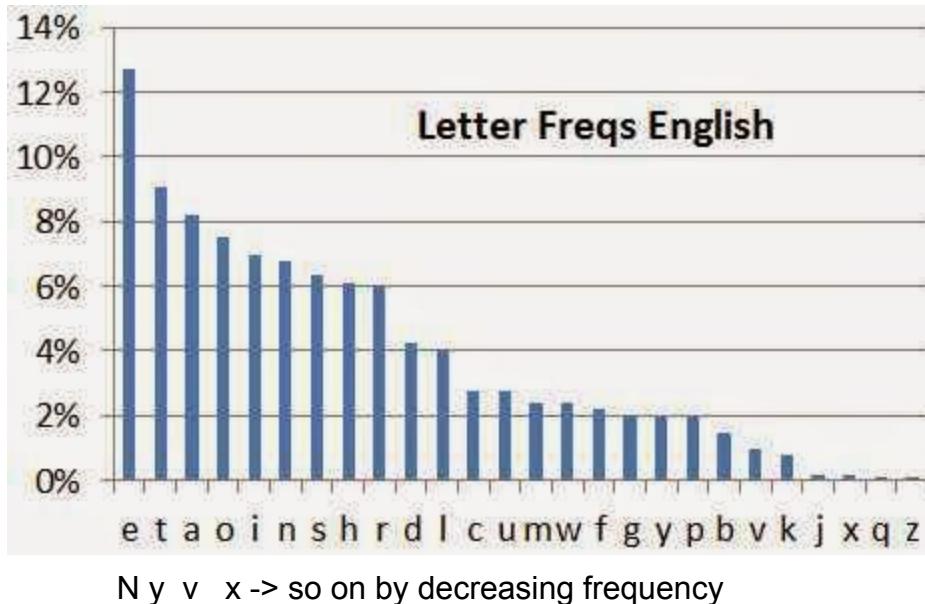
Lab Tasks

Task 1: Frequency Analysis Against Monoalphabetic Substitution Cipher.

First we have to find the frequency of every single letter that appears on the ciphertext.txt.

```
Terminal
[06/25/18] seed@VM:~/.../SOURCE CODE$ python task1.py
HERE IS THE FREQUENCY TABLES THAT READING FROM CIPHER TEXT
defaultdict(<type 'int'>, {' ': 741, 'a': 116, 'c': 104, 'b': 83, 'e': 76, 'd': 59, 'g': 83, 'f': 49, 'i': 166, 'h': 235, 'k': 5, 'j': 5, 'm': 264, 'l': 90, 'o': 4, 'n': 488, 'q': 276, 'p': 156, 's': 19, 'r': 82, 'u': 280, 't': 183, 'w': 1, 'v': 348, 'y': 373, 'x': 291, 'z': 95})
```

After that we sort the frequency letters by decreasing (except for \n) after that we found the list of letters:



After we mapping the letter with the cipher and checking for every single one by using “string maketrans python”: (note it did it in the long way to check one word, three words, four words)

For example first case will be N -> e because in the ciphertext.txt there is only one N letter in the sentence. Then we can see clearly that ‘v’ in cipher.txt is only by

guessing i assume that 'v -> a' . First three letters in ciphertext.txt is ytn so replace with n will be 'yte' the english word with e at the end can only be 'the' so replace t -> h and y -> t. Only looking for three letters word to finding the rest.

After all that replace i found the solution:

```
alphabet: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
key : c f m y p v b r l q x w i e j d s g k h n a z o t u
```

Making a python program that can output the plain text with the key and alphabet by using makestran syntax,

```
9 import collections  
10 import string  
11 from string import maketrans  
12  
13  
14  
15 # mapping function for substitution in the cipher text a-> c and b -> f so on |  
16  
17 alphabet = "abcdefghijklmnopqrstuvwxyz"  
18  
19 key = "cfmypvbrlqxwiejdsgkhnazotu"  
20  
21 transformer= string.maketrans(alphabet,key)  
22  
23  
24  
25 def frequency(text) :  
26     freq = collections.defaultdict(int)  
27     for line in text:  
28         line = line.rstrip('\n')  
29         for char in line:  
30             freq[char] +=1  
31  
32     return freq  
33  
34  
35 def printplaintext(text):  
36     plaintext =text.translate(transformer)  
37     return plaintext  
38  
39 def writeplaintext(text):  
40     with open('plaintext.txt', 'w') as f:  
41         f.write(text)
```

Result of plaintext.txt

```
ext.txt (~/Desktop/task_4/task1/SOURCE CODE) - gedit
Open Save
*task1.py x plaintext.txt x
1 the oscars turn on sunday which seems about right after this long strange
2 awards trip the bagger feels like a nonagenarian too
3
4 the awards race was bookended by the demise of harvey weinstein at its outset
5 and the apparent implosion of his film company at the end and it was shaped by
6 the emergence of metoo times up blackgown politics armchair 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 sported 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
```

Plain Text Tab Width: 8 Ln 1, Col 1 INS

Task 2: Encryption using Different Ciphers and Modes

I am using the same plaintext.txt that i got from first task

Encryption:

-aes-128-cbc

```
[06/22/18]seed@VM:~/Desktop$ openssl enc -aes-128-cbc
-e -in plain.txt -out cipher.bin \-k 123456 \-iv abc12
3456
```

-aes-128-cfb

```
[06/22/18]seed@VM:~/Desktop$ openssl enc -aes-128-cfb
-e -in plain.txt -out cipher.bin \-k 123456 \-iv abc12
3456
```

-bf-cbc

```
[06/22/18]seed@VM:~/Desktop$ openssl enc -bf-cbc -e -i  
n plain.txt -out cipher.bin \-k 123456 \-iv abc123456
```

Note: all encrypted files already list on the folder Name task 2.

Task 3: Encryption Mode – ECB vs. CBC

- List all possible cipher mode that can be used to encrypt or decrypt by using command "MAN ENC"

```
        aes-[128|192|256]-cbc 128/192/256 bit AES in  
CBC mode  
        aes-[128|192|256]           Alias for aes-[128|192|  
256]-cbc  
        aes-[128|192|256]-cfb 128/192/256 bit AES in  
128 bit CFB mode  
        aes-[128|192|256]-cfb1 128/192/256 bit AES in  
1 bit CFB mode  
        aes-[128|192|256]-cfb8 128/192/256 bit AES in  
8 bit CFB mode  
        aes-[128|192|256]-ecb 128/192/256 bit AES in  
ECB mode  
        aes-[128|192|256]-ofb 128/192/256 bit AES in
```

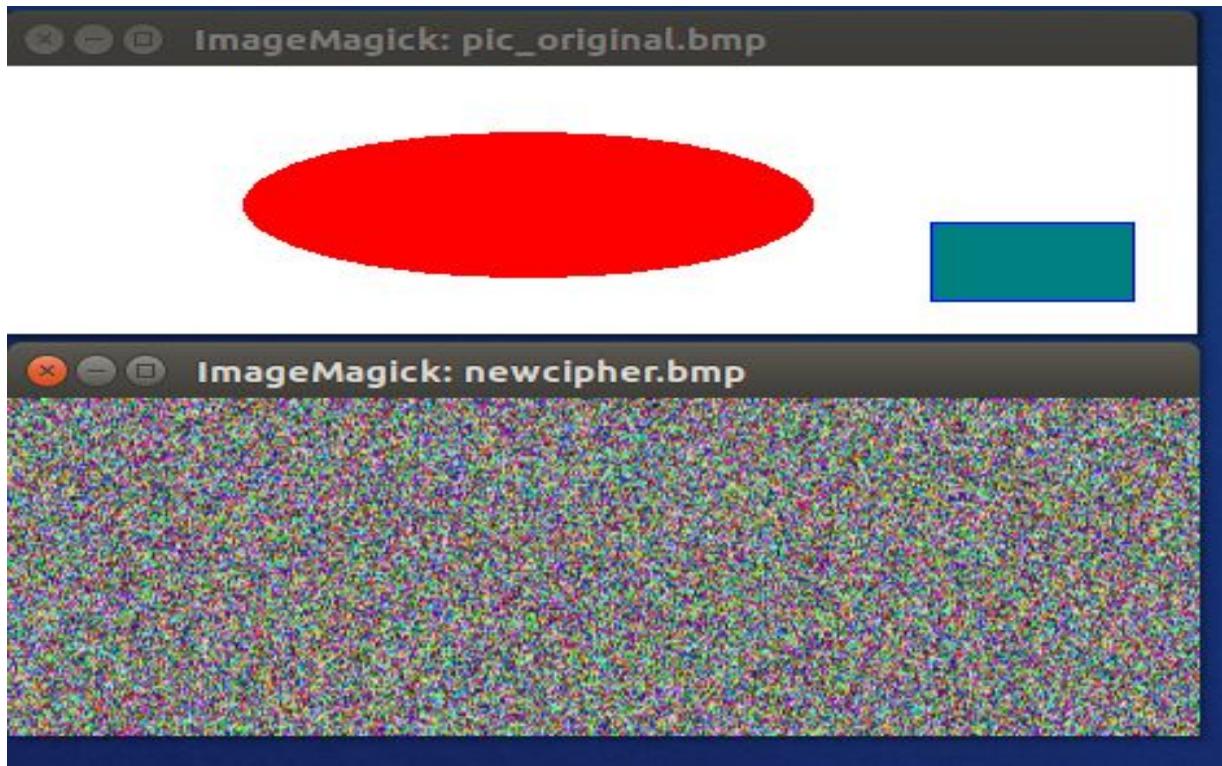
- Encryption the original pic given from the the description:

Mode: aes-128-cbc

Command : by using cat header and body we get header from original pic and combine with the data from encrypted pic so we can view the pic context.

```
[06/23/18]seed@VM:~/.../aes-128-cbc$ openssl enc -aes-128-cbc -e -in pic_original.bmp -out cipher.bmp \-k 123456 \-iv abc123456  
[06/23/18]seed@VM:~/.../aes-128-cbc$ head -c 54 pic_original.bmp >header  
[06/23/18]seed@VM:~/.../aes-128-cbc$ tail -c +55 cipher.bmp >body  
[06/23/18]seed@VM:~/.../aes-128-cbc$ cat header body > newcipher.bmp  
[06/23/18]seed@VM:~/.../aes-128-cbc$
```

Result : by seeing that we can see the original pic that encrypted to new cipher.bmp the picture turns to be a mess.

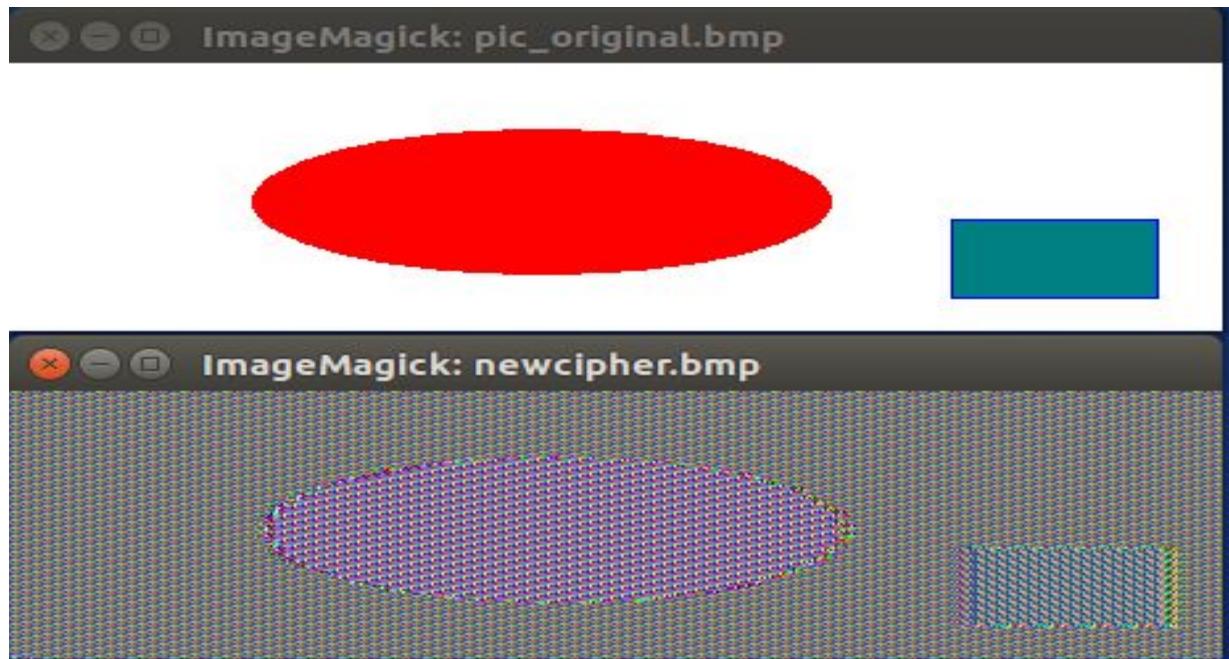


Mode: aes-128-ecb

command :

```
[06/23/18]seed@VM:~/.../encrypted original pic$ head -c 54 pic_original.bmp >header  
[06/23/18]seed@VM:~/.../encrypted original pic$ tail -c +55 cipher.bmp >body  
[06/23/18]seed@VM:~/.../encrypted original pic$ cat header body > newcipher.bmp  
[06/23/18]seed@VM:~/.../encrypted original pic$
```

Result: as we can see clearly that in ecb mode the pic generated not a mess anymore but it has some change about the color in the new pic.



- Doing the same step with a new bmp pic.

Mode: aes-128-cbc

```
[06/23/18]seed@VM:~/.../aes-128-cbc$ openssl enc -aes-128-cbc -e -in index.bmp -out cipher.bmp \-k 123456789aabcccd \-iv 0000000000000000  
[06/23/18]seed@VM:~/.../aes-128-cbc$ head -c 54 index.bmp >header  
[06/23/18]seed@VM:~/.../aes-128-cbc$ tail -c +55 cipher.bmp >body  
[06/23/18]seed@VM:~/.../aes-128-cbc$ cat header body > newcipher.bmp  
[06/23/18]seed@VM:~/.../aes-128-cbc$
```

result :



Mode: aes-128-ecb

```
[06/23/18]seed@VM:~/.../aes-128-ecb$ openssl enc -aes-128-ecb -e -in index.bmp -out cipher.bmp \-k 123456789aabcccd \-iv 0000000000000000  
warning: iv not use by this cipher  
[06/23/18]seed@VM:~/.../aes-128-ecb$ head -c 54 index.bmp >header  
[06/23/18]seed@VM:~/.../aes-128-ecb$ tail -c +55 cipher.bmp >body  
[06/23/18]seed@VM:~/.../aes-128-ecb$ cat header body > .bmp  
[06/23/18]seed@VM:~/.../aes-128-ecb$ cat header body > newcipher.bmp  
[06/23/18]seed@VM:~/.../aes-128-ecb$
```

*result : when turn into a new bmp pic the second ecb mode still turn into a mess again
I think this pixel of the picture is not good enough for seeing the clearly picture.*



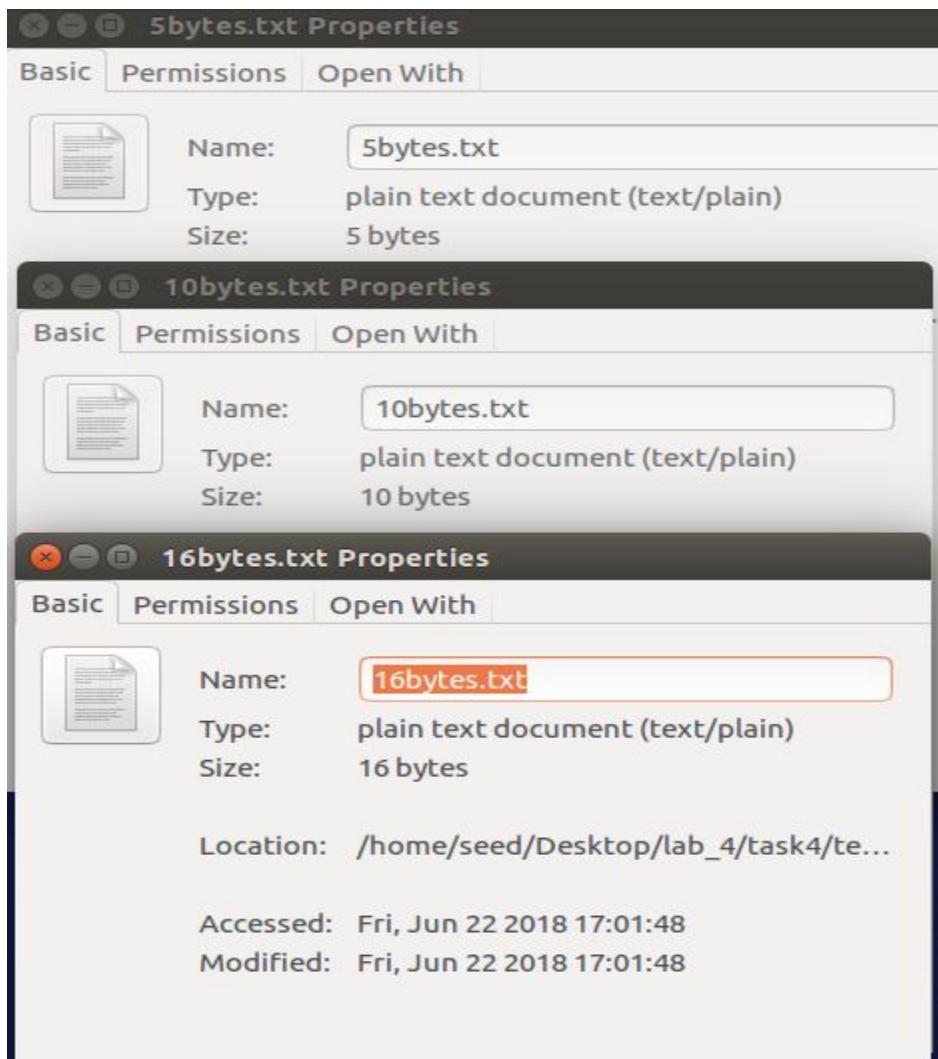
Task 4: Padding

- Creating three texts which difference sizes 5, 10 , 16 bytes.

Command:

```
[06/22/18]seed@VM:~/Desktop$ echo -n "12345" > 5bytes.txt
[06/22/18]seed@VM:~/Desktop$ echo -n "1234567891" > 10bytes.txt
[06/22/18]seed@VM:~/Desktop$ echo -n "1234567891012345" > 16bytes.txt
[06/22/18]seed@VM:~/Desktop$
```

Size of texts:



Hexdump command:

```
[06/24/18]seed@VM:~/.../text creating$ hexdump -C 5bytes.txt
00000000  31 32 33 34 35          |12345|
00000005
[06/24/18]seed@VM:~/.../text creating$ xxd 5bytes.txt
00000000: 3132 3334 35          12345
[06/24/18]seed@VM:~/.../text creating$ hexdump -C 10bytes.txt
00000000  31 32 33 34 35 36 37 38 39 31          |1234567891|
0000000a
[06/24/18]seed@VM:~/.../text creating$ xxd 10bytes.txt
00000000: 3132 3334 3536 3738 3931          1234567891
[06/24/18]seed@VM:~/.../text creating$ hexdump -C 16bytes.txt
00000000  31 32 33 34 35 36 37 38 39 31 30 31 32 33 34 35  |1234567891012345|
00000010
[06/24/18]seed@VM:~/.../text creating$ xxd 16bytes.txt
00000000: 3132 3334 3536 3738 3931 3031 3233 3435  1234567891012345
[06/24/18]seed@VM:~/.../text creating$ █
```

- Use ECB, CBC, CFB, and OFB modes to encrypt those texts:

Mode ECB:

Command:

```
[06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-ecb -e -in 16bytes.txt -out cipher16.bin \-k 00112233445566778889aabbccddeeff \-iv 0102030405060708
warning: iv not use by this cipher
[06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-ecb -e -in 10bytes.txt -out cipher10.bin \-k 00112233445566778889aabbccddeeff \-iv 0102030405060708
warning: iv not use by this cipher
[06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-ecb -e -in 5bytes.txt -out cipher5.bin \-k 00112233445566778889aabbccddeeff \-iv 0102030405060708
warning: iv not use by this cipher
[06/24/18]seed@VM:~/.../text creating$
```

Size after encrypted:

```
[06/24/18]seed@VM:~/.../ecb$ ls -l
total 16
-rw-rw-r-- 1 seed seed  32 Jun 24 03:01 cipher10.bin
-rw-rw-r-- 1 seed seed  48 Jun 24 03:01 cipher16.bin
-rw-rw-r-- 1 seed seed  32 Jun 24 03:02 cipher5.bin
```

Hexdump:

```
[06/24/18]seed@VM:~/.../ecb$ hexdump -C cipher5.bin
00000000  53 61 6c 74 65 64 5f 5f  2b 8a 5a 53 c5 ba 3b 85  |Salted_+.ZS..;.| 
00000010  cb 12 d9 cc 1c e1 cf a5  66 2b c8 2b 12 c3 a9 df  |.....f+.+....| 
00000020

[06/24/18]seed@VM:~/.../ecb$ xxd cipher5.bin
00000000: 5361 6c74 6564 5f5f 2b8a 5a53 c5ba 3b85  Salted_+.ZS..;.
00000010: cb12 d9cc 1ce1 cfa5 662b c82b 12c3 a9df  .....f+.+.....
[06/24/18]seed@VM:~/.../ecb$ hexdump -C cipher10.bin
00000000  53 61 6c 74 65 64 5f 5f  e4 d7 00 fa 03 7f 86 9b  |Salted_.....|
00000010  ac 15 43 4d 00 a6 e1 31  9d 32 48 a8 ee a5 09 f2  |..CM...1.2H.....|
00000020

[06/24/18]seed@VM:~/.../ecb$ xxd cipher10.bin
00000000: 5361 6c74 6564 5f5f e4d7 00fa 037f 869b  Salted_.....
00000010: ac15 434d 00a6 e131 9d32 48a8 eea5 09f2  ..CM...1.2H.....
[06/24/18]seed@VM:~/.../ecb$ xxd cipher16.bin
00000000: 5361 6c74 6564 5f5f f864 d047 8f5c 34df  Salted_.d.G.\4.
00000010: 805a 1734 4f1e c831 160f 0831 c056 57da  .Z.40..1...1.VW.
00000020: 04c0 c82c 110e c9b7 8886 729d dcf5 c14e  ....,...,r....N
[06/24/18]seed@VM:~/.../ecb$ hexdump -C cipher16.bin
00000000  53 61 6c 74 65 64 5f 5f  f8 64 d0 47 8f 5c 34 df  |Salted_.d.G.\4.|
00000010  80 5a 17 34 4f 1e c8 31  16 0f 08 31 c0 56 57 da  |.Z.40..1...1.VW.|
00000020  04 c0 c8 2c 11 0e c9 b7  88 86 72 9d dc f5 c1 4e  |...,,...,r....N|
00000030

[06/24/18]seed@VM:~/.../ecb$
```

Mode CBC:

- *Command:*

```
06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-cbc -e -in 5bytes.txt -out cipher5.bin \-k 00112233445566778889aabcccddeeff \-iv 0102030405060708
06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-cbc -e -in 10bytes.txt -out cipher10.bin \-k 00112233445566778889aabcccddeeff \-iv 01020304050608
06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-cbc -e -in 16bytes.txt -out cipher16.bin \-k 00112233445566778889aabcccddeeff \-iv 01020304050608
06/24/18]seed@VM:~/.../text creating$
```

- *Size :*

```
[06/24/18]seed@VM:~/.../cbc$ ls -l
total 16
-rw-rw-r-- 1 seed seed 32 Jun 24 02:56 cipher10.bin
-rw-rw-r-- 1 seed seed 48 Jun 24 02:56 cipher16.bin
-rw-rw-r-- 1 seed seed 32 Jun 24 02:54 cipher5.bin
```

-

- *hexdump :*

```

Terminal
[06/24/18]seed@VM:~/.../cbc$ hexdump -C cipher5.bin
00000000  53 61 6c 74 65 64 5f 5f  7a f9 63 c0 fb 64 34 43  |Salted_z.c..d4C|
00000010  ab a8 c4 3b ff 97 a4 90  87 02 c9 ee 33 18 fe 8d  |....;.....3...|
00000020
[06/24/18]seed@VM:~/.../cbc$ xxd cipher5.bin
00000000: 5361 6c74 6564 5f5f 7af9 63c0 fb64 3443 Salted_z.c..d4C
00000010: aba8 c43b ff97 a490 8702 c9ee 3318 fe8d ...;.....3...
[06/24/18]seed@VM:~/.../cbc$ hexdump -C cipher10.bin
00000000  53 61 6c 74 65 64 5f 5f  70 39 04 59 5c 91 7a b0  |Salted_p9.Y\.z.|
00000010  ba 26 a0 bd 1a 26 48 4f  fa 1d ce 5f 71 c3 fb f6  |.&...&H0..._q...|
00000020
[06/24/18]seed@VM:~/.../cbc$ xxd cipher10.bin
00000000: 5361 6c74 6564 5f5f 7039 0459 5c91 7ab0 Salted_p9.Y\.z.
00000010: ba26 a0bd 1a26 484f fa1d ce5f 71c3 fbf6 .&...&H0..._q...
[06/24/18]seed@VM:~/.../cbc$ hexdump -C cipher16.bin
00000000  53 61 6c 74 65 64 5f 5f  43 54 4f 20 e5 90 d8 2f  |Salted_CTO .../|
00000010  a0 c7 1d 76 d5 75 2b 38  ba 62 06 26 88 5b 00 4c  |...v.u+8.b.&.[.L|
00000020  3a d0 1f 22 09 12 39 4d  bf ee d4 85 fa 4d 91 dd  |...".9M....M..|
00000030
[06/24/18]seed@VM:~/.../cbc$ xxd cipher16.bin
00000000: 5361 6c74 6564 5f5f 4354 4f20 e590 d82f Salted_CTO ...
00000010: a0c7 1d76 d575 2b38 ba62 0626 885b 004c ...v.u+8.b.&.[.L
00000020: 3ad0 1f22 0912 394d bfee d485 fa4d 91dd ...".9M....M..
[06/24/18]seed@VM:~/.../cbc$ █

```

Mode CFB:

- *Command:*

```

Terminal
[06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-cfb -e -in 5bytes.txt -out cipher5.bin \-k 00112233445566778889aabccddeeff \-iv 0102030405060708
[06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-cfb -e -in 10bytes.txt -out cipher10.bin \-k 00112233445566778889aabccddeeff \-iv 0102030405060708
[06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-cfb -e -in 16bytes.txt -out cipher16.bin \-k 00112233445566778889aabccddeeff \-iv 0102030405060708

```

- *Size :*

```

Terminal
[06/24/18]seed@VM:~/.../cfb$ ls -l
total 16
-rw-rw-r-- 1 seed seed   26 Jun 24 02:59 cipher10.bin
-rw-rw-r-- 1 seed seed   32 Jun 24 03:00 cipher16.bin
-rw-rw-r-- 1 seed seed   21 Jun 24 02:59 cipher5.bin

```

- *hexdump* :

```
[06/24/18]seed@VM:~/.../cfb$ hexdump -C cipher5.bin
00000000  53 61 6c 74 65 64 5f 5f  84 58 c5 e8 74 8a 27 1a  |Salted__.X..t.'.| 
00000010  18 b7 b4 48 45                                     |...HE|
00000015
[06/24/18]seed@VM:~/.../cfb$ xxd cipher5.bin
00000000: 5361 6c74 6564 5f5f 8458 c5e8 748a 271a  Salted__.X..t.'.
00000010: 18b7 b448 45                                     ...HE
[06/24/18]seed@VM:~/.../cfb$ hexdump -C cipher10.bin
00000000  53 61 6c 74 65 64 5f 5f  a7 b0 d9 b2 4b 2b e3 7c  |Salted__...K+.|| 
00000010  27 97 27 58 e5 be d0 c5  7a aa                     |'.'X....z.| 
0000001a
[06/24/18]seed@VM:~/.../cfb$ xxd cipher10.bin
00000000: 5361 6c74 6564 5f5f a7b0 d9b2 4b2b e37c  Salted__...K+.|
00000010: 2797 2758 e5be d0c5 7aaa                     .'.'X....z.
[06/24/18]seed@VM:~/.../cfb$ hexdump -C cipher16.bin
00000000  53 61 6c 74 65 64 5f 5f  fa f5 3c 5a b1 b4 2d e9  |Salted__..<Z...| 
00000010  f2 a7 d6 07 0b 4c f9 0b  c2 e8 c1 a5 43 68 11 dd  |....L.....Ch..| 
00000020
[06/24/18]seed@VM:~/.../cfb$ xxd cipher16.bin
00000000: 5361 6c74 6564 5f5ffaf5 3c5a b1b4 2de9  Salted__..<Z... .
00000010: f2a7 d607 0b4c f90b c2e8 c1a5 4368 11dd  ....L.....Ch..
[06/24/18]seed@VM:~/.../cfb$
```

Mode OFB:

- *Command:*

```
[06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-ofb -e -in 5bytes.txt -out cipher5.bin \-k 00112233445566778889aabccddeeff \-iv 0102030405060708
[06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-ofb -e -in 10bytes.txt -out cipher10.bin \-k 00112233445566778889aabccddeeff \-iv 0102030405060708
[06/24/18]seed@VM:~/.../text creating$ openssl enc -aes-128-ofb -e -in 16bytes.txt -out cipher16.bin \-k 00112233445566778889aabccddeeff \-iv 0102030405060708
[06/24/18]seed@VM:~/.../text creating$
```

- *Size :*

```
[06/24/18]seed@VM:~/.../ofb$ ls -l
total 16
-rw-rw-r-- 1 seed seed    26 Jun 24 03:03 cipher10.bin
-rw-rw-r-- 1 seed seed    32 Jun 24 03:03 cipher16.bin
-rw-rw-r-- 1 seed seed    21 Jun 24 03:03 cipher5.bin
```

- *hexdump :*

```

Terminal
[06/24/18]seed@VM:~/.../ofb$ hexdump -C cipher5.bin
00000000  53 61 6c 74 65 64 5f 5f  f4 80 74 88 e5 af db d0  |Salted___.t....|
00000010  b7 dd 8b d2 fd
00000015
[06/24/18]seed@VM:~/.../ofb$ xxd cipher5.bin
00000000: 5361 6c74 6564 5f5f f480 7488 e5af dbd0  Salted___.t.....
00000010: b7dd 8bd2 fd      .....
[06/24/18]seed@VM:~/.../ofb$ hexdump -C cipher10.bin
00000000  53 61 6c 74 65 64 5f 5f  bc 9b da c6 23 d8 ee 6f  |Salted___.#..o|
00000010  94 dd 5a 6d af 62 34 f3  14 51                  |..Zm.b4..Q|
0000001a
[06/24/18]seed@VM:~/.../ofb$ xxd cipher10.bin
00000000: 5361 6c74 6564 5f5f bc9b dac6 23d8 ee6f  Salted___.#..o
00000010: 94dd 5a6d af62 34f3 1451                  ..Zm.b4..Q
[06/24/18]seed@VM:~/.../ofb$ hexdump -C cipher16.bin
00000000  53 61 6c 74 65 64 5f 5f  06 85 65 07 e5 7b e8 12  |Salted___.e..{..|
00000010  15 d3 5b 0a 16 fd 28 ea  00 a0 c6 a9 22 06 b5 7d  |...[...{....."}]| 
00000020
[06/24/18]seed@VM:~/.../ofb$ xxd cipher16.bin
00000000: 5361 6c74 6564 5f5f 0685 6507 e57b e812  Salted___.e..{..
00000010: 15d3 5b0a 16fd 28ea 00a0 c6a9 2206 b57d  ...[...{....."}]
[06/24/18]seed@VM:~/.../ofb$ █

```

Conclusion:

- By default all cipher mode will add 16 bytes iv for encryption but in CBC and ECB will add another padding size for every single one file that encrypted by those mode. Those mode are Blocks cipher so those need to fill every single block with 16 bits

In CBC mode:

5 bytes -> 32 bytes | 32 - 5 = 27 bytes -> 27 - 16 bytes iv = 11 byte padding
 10 bytes -> 32 bytes | 6 bytes padding
 16 bytes -> 48 bytes | 16 bytes padding

In ECB mode:

5 bytes -> 32 bytes | 11 byte padding
 10 bytes -> 32 bytes | 6 bytes padding
 16 bytes -> 48 bytes | 16 bytes padding

- In CFB and OFB mode we don't need to add anything padding because those cipher mode are stream cipher:

In CFB mode:

5 bytes -> 21 bytes | 21 - 5 = 16 bytes -> 16 - 16 = 0 bytes padding

10 bytes -> 26 bytes | 0

16 bytes -> 32 bytes | 0

In OFB mode:

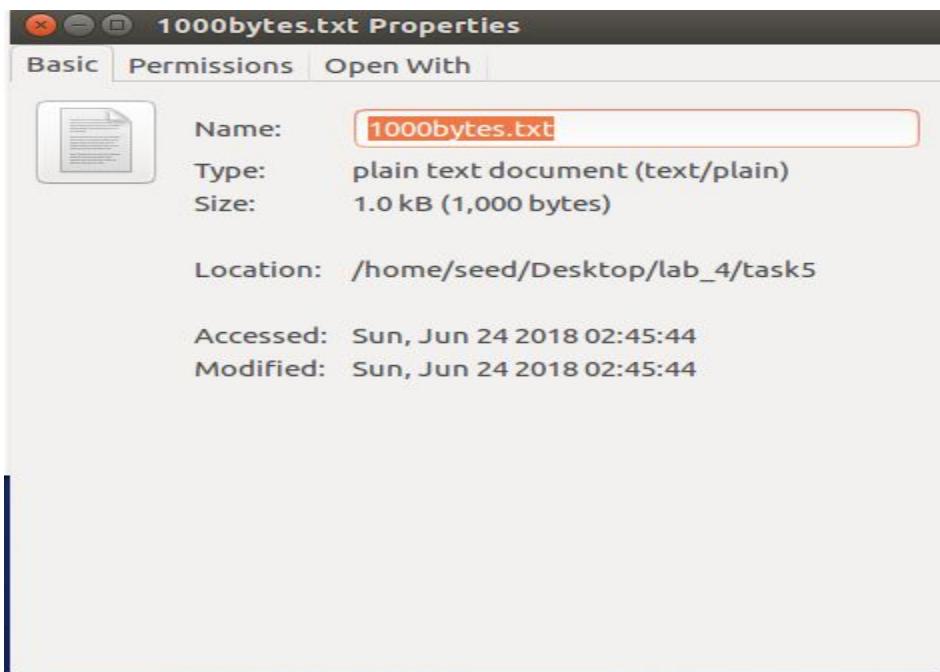
5 bytes -> 21 bytes | 0

10 bytes -> 26 bytes | 0

16 bytes -> 32 bytes | 0

Task 5: Error Propagation – Corrupted CipherText

- Creating 1000 bytes plaintext



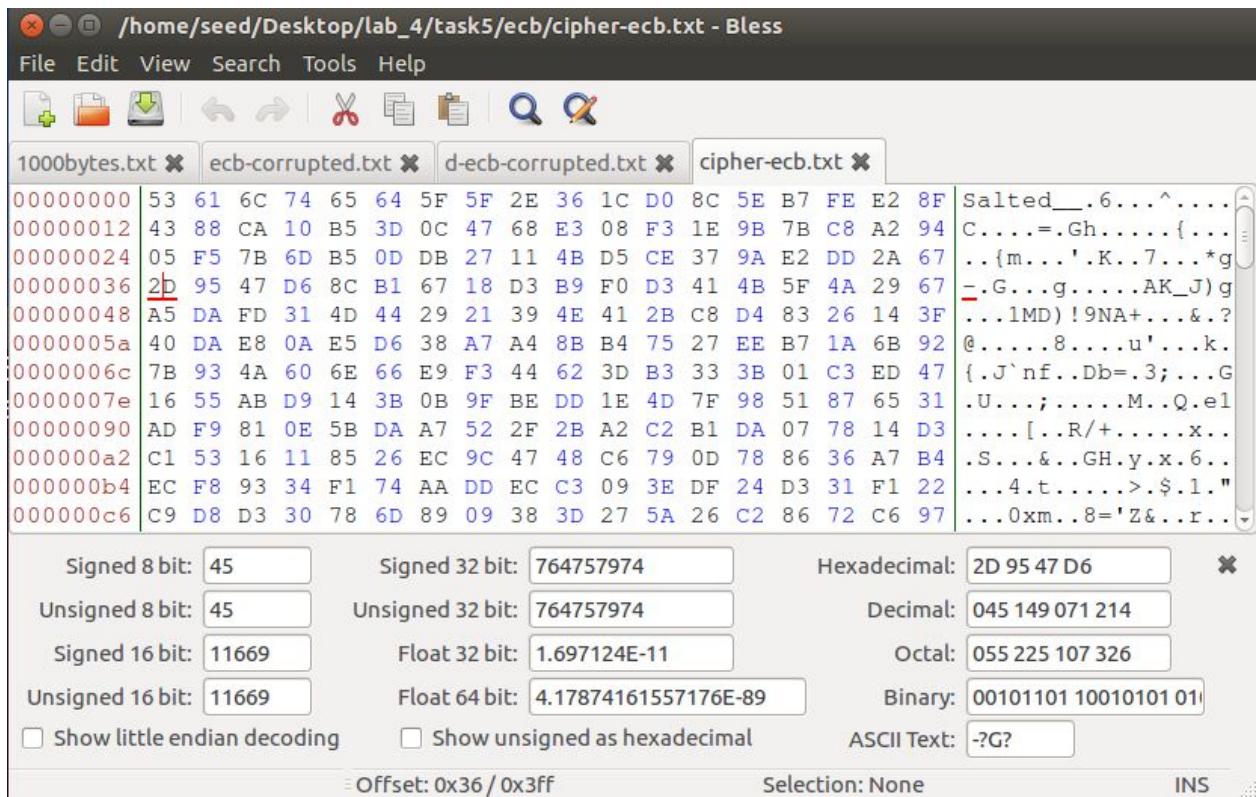
- Encrypted with 4 mode cipher: ECB, CBC, CFB, or OFB, respectively

```
[06/24/18]seed@VM:~/.../task5$ openssl enc -aes-128-cbc -e -in 1000bytes.txt -out cipher-cbc.txt \-k 00112233445566778889aabbccddeeff \-iv 0102030405060708
[06/24/18]seed@VM:~/.../task5$ openssl enc -aes-128-cfb -e -in 1000bytes.txt -out cipher-cfb.txt \-k 00112233445566778889aabbccddeeff \-iv 0102030405060708
[06/24/18]seed@VM:~/.../task5$ openssl enc -aes-128-ecb -e -in 1000bytes.txt -out cipher-ecb.txt \-k 00112233445566778889aabbccddeeff \-iv 0102030405060708
warning: iv not use by this cipher
[06/24/18]seed@VM:~/.../task5$ openssl enc -aes-128-ofb -e -in 1000bytes.txt -out cipher-ofb.txt \-k 00112233445566778889aabbccddeeff \-iv 0102030405060708
[06/24/18]seed@VM:~/.../task5$
```

MODE ECB:

HEX command to view cipher block cycles

- Original cipher



- Corrupted cipher (by changing up one bit at position 55th)

The screenshot shows the Bless hex editor interface. At the top, there are tabs for four files: 1000bytes.txt, ecb-corrupted.txt, d-ecb-corrupted.txt, and cipher-ecb.txt. The ecb-corrupted.txt tab is active, showing a hex dump of the corrupted ECB cipher. A specific byte at address 0x36 (hex 2E) has been modified from its original value of 95 to 94. Below the hex dump, there is a set of conversion tools:

Signed 8 bit: 46	Signed 32 bit: 781535190	Hexadecimal: 2E 95 47 D6
Unsigned 8 bit: 46	Unsigned 32 bit: 781535190	Decimal: 046 149 071 214
Signed 16 bit: 11925	Float 32 bit: 6.788496E-11	Octal: 056 225 107 326
Unsigned 16 bit: 11925	Float 64 bit: 2.73858010518111E-84	Binary: 00101110 10010101 011

There are also checkboxes for "Show little endian decoding" and "Show unsigned as hexadecimal". At the bottom, there are fields for "Offset: 0x36 / 0x3ff", "Selection: None", and an "INS" button.

- Decrypted the corrupted ciphertext

```
[06/24/18]seed@VM:~/.../ecb$ openssl enc -aes-128-ecb -d -in ecb-corrupted.txt -out d-ecb-corrupted.txt \-k 00112233445566778889aabbccddeeff \-iv 0102030405060708
warning: iv not use by this cipher
```

The screenshot shows the Bless hex editor interface again. The d-ecb-corrupted.txt tab is active, displaying the decrypted ECB cipher. The byte at address 0x36 (hex 2E) has been restored to its original value of 95. Below the hex dump, there is a set of conversion tools:

Signed 8 bit: 101	Signed 32 bit: 1701978228	Hexadecimal: 65 72 20 74
Unsigned 8 bit: 101	Unsigned 32 bit: 1701978228	Decimal: 101 114 032 116
Signed 16 bit: 25970	Float 32 bit: 7.146321E+22	Octal: 145 162 040 164
Unsigned 16 bit: 25970	Float 64 bit: 4.70108376461316E+180	Binary: 01100101 01110010 00

There are also checkboxes for "Show little endian decoding" and "Show unsigned as hexadecimal". At the bottom, there are fields for "Offset: 0x36 / 0x3e7", "Selection: None", and an "INS" button.

- Result

```
the oscars turn on sunday which [redacted] -2ji  
61ht 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 seem eKtra long it was  
eKtra 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 Juestion surrounding this years academy awards is how or if the  
ceremony will address metoo especially after the golden globes which became  
a Oubilant comingout party for times up the movement spearheaded by  
powerful hollywood women who helped raise millions of dollars to fight seKual  
harassment around the count
```

MODE CBC:

HEX command to view cipher block cycles:

- Original cipher

Signed 8 bit:	77	Signed 32 bit:	1299966995	Hexadecimal:	4D 7B EC 13
Unsigned 8 bit:	77	Unsigned 32 bit:	1299966995	Decimal:	077 123 236 019
Signed 16 bit:	19835	Float 32 bit:	2.641595E+08	Octal:	115 173 354 023
Unsigned 16 bit:	19835	Float 64 bit:	1.83784295774991E+65	Binary:	01001101 01111011 11
<input type="checkbox"/> Show little endian decoding		<input type="checkbox"/> Show unsigned as hexadecimal		ASCII Text:	M{? [redacted]}

Offset: 0x36 / 0x3ff Selection: None INS

- Corrupted cipher (by changing up one bit at position 55th)

The screenshot shows the Bless hex editor interface. The file 'cbc-corrupted' is open. The hex dump shows several bytes being modified. At offset 36, the byte 7B is changed to 7A. The ASCII text view shows the original message 'Salted__z..B..Y8~...' followed by corrupted data. Below the hex dump, conversion tools are available for various data types.

Signed 8 bit:	123	Signed 32 bit:	2079069175	Hexadecimal:	7B EC 13 F7
Unsigned 8 bit:	123	Unsigned 32 bit:	2079069175	Decimal:	123 236 019 247
Signed 16 bit:	31724	Float 32 bit:	2.451574E+36	Octal:	173 354 023 367
Unsigned 16 bit:	31724	Float 64 bit:	8.55090479793048E+288	Binary:	01111011 11101100 00

Show little endian decoding Show unsigned as hexadecimal ASCII Text: {? 00 13 ?}

Offset: 0x37 / 0x3ff Selection: None INS

- Decrypted the corrupted ciphertext

The screenshot shows the Bless hex editor interface. The file 'd-cbc-corrupted.txt' is open, displaying the decrypted message. The message starts with 'the oscars turn o...' and continues with several lines of text. The original file 'cbc-corrupted' is listed in the tabs. Below the hex dump, conversion tools are available for various data types.

Signed 8 bit:	102	Signed 32 bit:	1718755444	Hexadecimal:	66 72 20 74
Unsigned 8 bit:	102	Unsigned 32 bit:	1718755444	Decimal:	102 114 032 116
Signed 16 bit:	26226	Float 32 bit:	2.858528E+23	Octal:	146 162 040 164
Unsigned 16 bit:	26226	Float 64 bit:	3.08090225597688E+185	Binary:	01100110 01110010 00

Show little endian decoding Show unsigned as hexadecimal ASCII Text: fr t

Offset: 0x36 / 0x3e7 Selection: None INS

- result

```
the oscars turn on sunday which^á÷. | ^{ÚøpN5b Hht aftfr 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 seem eKtra long it was eKtra 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 Juestion surrounding this years academy awards is how or if the ceremony will address metoo especially after the golden globes which became a Oubilant comingout party for times up the movement spearheaded by powerful hollywood women who helped raise millions of dollars to fight seKual harassment around the count
```

MODE CFB:

HEX command to view cipher block cycles:

- Original cipher

Signed 8 bit:	76	Signed 32 bit:	1287592364	Hexadecimal:	4C BF 19 AC
Unsigned 8 bit:	76	Unsigned 32 bit:	1287592364	Decimal:	076 191 025 172
Signed 16 bit:	19647	Float 32 bit:	1.001916E+08	Octal:	114 277 031 254
Unsigned 16 bit:	19647	Float 64 bit:	4.99762302788904E+61	Binary:	01001100 10111111 00

Show little endian decoding Show unsigned as hexadecimal ASCII Text: L? 19 ?

Offset: 0x36 / 0x3f7 Selection: None INS

- Corrupted cipher (by changing up one bit at position 55th)

The screenshot shows the Bless hex editor interface. At the top, there are tabs for four files: 1000bytes.txt, cfb-corrupted.txt, d-cfb-corrupted.txt, and cipher-cfb.txt. The d-cfb-corrupted.txt tab is active, displaying a corrupted AES-128-CFB ciphertext. Below the editor, there are conversion tools for different data types:

Signed 8 bit:	77	Signed 32 bit:	1304369580	Hexadecimal:	4D BF 19 AC
Unsigned 8 bit:	77	Unsigned 32 bit:	1304369580	Decimal:	077 191 025 172
Signed 16 bit:	19903	Float 32 bit:	4.007663E+08	Octal:	115 277 031 254
Unsigned 16 bit:	19903	Float 64 bit:	3.27524222755736E+66	Binary:	01001101 10111111 00
<input type="checkbox"/> Show little endian decoding		<input type="checkbox"/> Show unsigned as hexadecimal		ASCII Text:	M?@?

At the bottom, there are offset and selection fields, and an 'INS' button.

- Decrypted the corrupted ciphertext

The terminal window shows the command:

```
[06/24/18] seed@VM:~/.../cfb$ openssl enc -aes-128-cfb -d -in cfb-corrupted.txt -out d-cfb-corrupted.txt \-k 00112233445566778899aabbccddeeff \-iv 0102030405060708
```

The Bless hex editor shows the decrypted AES-128-CFB ciphertext. The corruption from the previous screenshot has been removed, resulting in readable English text.

Signed 8 bit:	-56	Signed 32 bit:	-936009643	Hexadecimal:	C8 35 A0 55
Unsigned 8 bit:	200	Unsigned 32 bit:	3358957653	Decimal:	200 053 160 085
Signed 16 bit:	-14283	Float 32 bit:	-185985.3	Octal:	310 065 240 125
Unsigned 16 bit:	51253	Float 64 bit:	-7.35904774633702E+39	Binary:	11001000 00110101 10
<input type="checkbox"/> Show little endian decoding		<input type="checkbox"/> Show unsigned as hexadecimal		ASCII Text:	?5?U

At the bottom, there are offset and selection fields, and an 'INS' button.

- result

```

the oscars turn on sunday which seems!about rig&fp VÝÈS U
``xRH[ing 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 oust seem ektra long it was
ektra 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 Juestion surrounding this years academy awards is how or if the
ceremony will address metoo especially after the golden globes which became
a Oubilant comingout party for times up the movement spearheaded by
powerful hollywood women who helped raise millions of dollars to fight seKual
harassment around the count

```

MODE OFB:

HEX command to view cipher block cycles:

- Original cipher

Signed 8 bit: -10	Signed 32 bit: -160342883	Hexadecimal: F6 71 5C 9D
Unsigned 8 bit: 246	Unsigned 32 bit: 4134624413	Decimal: 246 113 092 157
Signed 16 bit: -2447	Float 32 bit: -1.22385E+33	Octal: 366 161 134 235
Unsigned 16 bit: 63089	Float 64 bit: -3.41688503194275E+262	Binary: 11110110 01110001 01
<input type="checkbox"/> Show little endian decoding	<input type="checkbox"/> Show unsigned as hexadecimal	ASCII Text: ?q\?

Offset: 0x36 / 0x3f7 Selection: None INS

- Corrupted cipher (by changing up one bit at position 55th)

```

00000000 53 61 6C 74 65 64 5F 5F 81 EF 21 AE C5 5E BD 3E B2 1A Salted_...!...^.>...
00000012 58 B5 1E 46 92 8B 9C 0E 29 33 58 7F 9B 99 27 49 64 05 X..F....)3X...'Id...
00000024 24 14 7B 52 66 6E CB 66 98 AF B2 BC B9 77 6C 0B FB 80 ${.Rfn.f....wl...
00000036 F7 71 5C 9D 4D 8C 46 2E 76 FF 6D 8C 11 B0 56 66 D2 C6 .g\..M.F.v.m...Vf...
00000048 17 81 94 12 26 C8 C4 01 8B B0 03 A7 72 24 56 F1 03 E0 ....&.....r$V...
0000005a 3B BD 2D 7D 5F 90 32 0A C9 0F 82 DC 7F 9D AE 74 FA EB ;.-}_.2....t...
0000006c DB 84 34 D5 C3 55 3C 0A 85 86 9E 81 1D C3 4B 8D 45 84 ..4..U<.....K.E.
0000007e 42 89 1A 63 00 B7 5C 67 94 BB C6 A3 94 BB 55 CD 33 01 B..c..\g.....U.3.
00000090 4E E2 25 07 2A 58 E1 1B 97 CD E0 D1 2E 56 E8 CF D7 34 N.%.*X.....V..4
000000a2 1F 8D 8E 02 BD 6A 38 81 44 A3 1E 4D D2 FB 3E A5 BD 3C .....j8.D..M..>..<
000000b4 51 DF 1E D0 31 12 70 01 E5 C0 41 66 4C 83 20 88 87 E8 Q...1.p...AfL. ...
000000c6 90 B2 87 12 EB 31 03 16 0A 1B 25 2F BC EB 33 9A 9E 1F .....1....%/.3...

```

Signed 8 bit: 113 Signed 32 bit: 1901894989 Hexadecimal: 71 5C 9D 4D
 Unsigned 8 bit: 113 Unsigned 32 bit: 1901894989 Decimal: 113 092 157 077
 Signed 16 bit: 29020 Float 32 bit: 1.09243E+30 Octal: 161 134 235 115
 Unsigned 16 bit: 29020 Float 64 bit: 1.16456092850852E+238 Binary: 01110001 01011100 10
 Show little endian decoding Show unsigned as hexadecimal ASCII Text: q\?M
 Offset: 0x37 / 0x3f7 Selection: None INS

- Decrypted the corrupted ciphertext

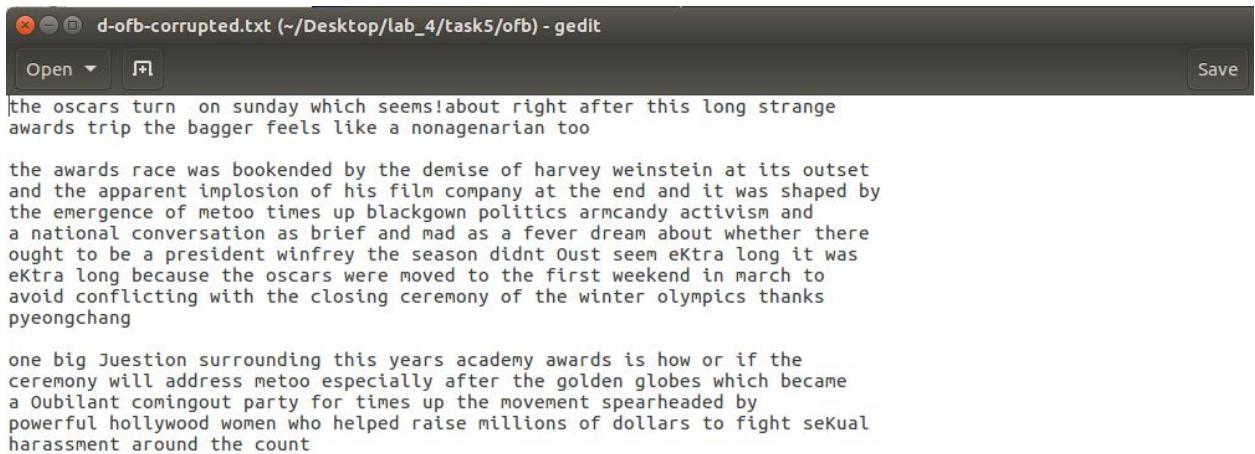
```

00000000 74 68 65 20 6F 73 63 61 72 73 20 74 75 72 6E 20 20 6F the oscars turn o
00000012 6E 20 73 75 6E 64 61 79 20 77 68 69 63 68 20 73 65 65 n sunday which see...
00000024 6D 73 21 61 62 6F 75 74 20 72 69 67 68 74 20 61 66 74 ms!about right aft...
00000036 65 72 20 74 68 69 73 20 6C 6F 6E 67 20 73 74 72 61 6E er this long stran...
00000048 67 65 0A 61 77 61 72 64 73 20 74 72 69 70 20 74 68 65 ge.awards trip the...
0000005a 20 62 61 67 67 65 72 20 66 65 65 6C 73 20 6C 69 6B 65 bagger feels like...
0000006c 20 61 20 6E 6F 6E 61 67 65 6E 61 72 69 61 6E 20 74 6F a nonagenarian to...
0000007e 6F 0A 0A 74 68 65 20 61 77 61 72 64 73 20 72 61 63 65 o..the awards race...
00000090 20 77 61 73 20 62 6F 6F 6B 65 6E 64 65 64 20 62 79 20 was bookended by...
000000a2 74 68 65 20 64 65 6D 69 73 65 20 6F 66 20 68 61 72 76 the demise of harv...
000000b4 65 79 20 77 65 69 6E 73 74 65 69 6E 20 61 74 20 69 74 ey weinstein at it...
000000c6 73 20 6F 75 74 73 65 74 0A 61 6E 64 20 74 68 65 20 61 s outset.and the a...

```

Signed 8 bit: 101 Signed 32 bit: 1701978228 Hexadecimal: 65 72 20 74
 Unsigned 8 bit: 101 Unsigned 32 bit: 1701978228 Decimal: 101 114 032 116
 Signed 16 bit: 25970 Float 32 bit: 7.146321E+22 Octal: 145 162 040 164
 Unsigned 16 bit: 25970 Float 64 bit: 4.70108376461316E+180 Binary: 01100101 01110010 00
 Show little endian decoding Show unsigned as hexadecimal ASCII Text: er t
 Offset: 0x36 / 0x3e7 Selection: None INS

- result



The screenshot shows a terminal window titled "d-ofb-corrupted.txt (~/Desktop/lab_4/task5/ofb) - gedit". The file contains two blocks of text. The first block is a single sentence: "the oscars turn on sunday which seems!about right after this long strange awards trip the bagger feels like a nonagenarian too". The second block is a multi-line paragraph about the awards race being shaped by the demise of Harvey Weinstein, the emergence of MeToo, and a national conversation about sexual harassment. It ends with "pyeongchang". The third block is another multi-line paragraph about the Academy Awards addressing MeToo issues, mentioning the Golden Globes and powerful Hollywood women raising millions for sexual harassment causes.

```
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 oust seem ektra long it was
ektra 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 count
```

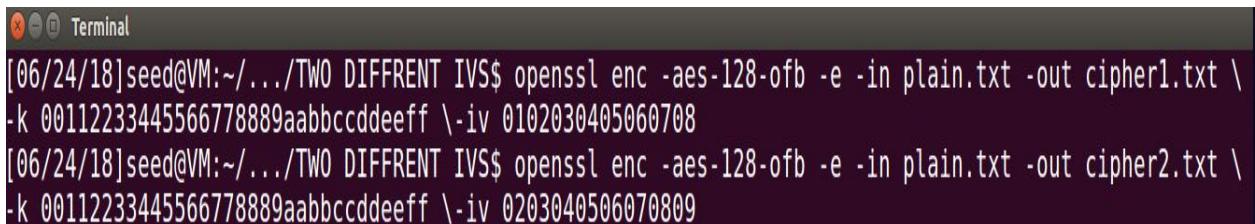
Conclusion: ECB, CBC, CFB, or OFB, mode

As the 4 result we can see clearly that the OFB, ECB, CBC, and CFB mode. We interpreted the 55th bit up to one the decryption will mess up the context of original text in 55th block cycles.

Task 6: Initial Vector (IV)

- Task 6.1. A basic requirement for IV is uniqueness:

- a) Two different IVs:



The screenshot shows a terminal window titled "Terminal". It displays two separate commands run on the same VM. Both commands use the "openssl enc" command with the "-aes-128-ofb" cipher, "-e" encryption mode, and "-in plain.txt" input. The first command uses "-out cipher1.txt" and has an IV of "0102030405060708". The second command also uses "-out cipher2.txt" and has an IV of "0203040506070809".

```
[06/24/18]seed@VM:~/.../TWO DIFFRENT IV$ openssl enc -aes-128-ofb -e -in plain.txt -out cipher1.txt \
-k 0011223344556677889aabcccddeeff \-iv 0102030405060708
[06/24/18]seed@VM:~/.../TWO DIFFRENT IV$ openssl enc -aes-128-ofb -e -in plain.txt -out cipher2.txt \
-k 0011223344556677889aabcccddeeff \-iv 0203040506070809
```

Result from cipher1.txt:

```
cipher1.txt (~/Desktop/lab_4/task6/TASK 6.1/TWO DIFFRENT IVS) - gedit
Open + Save
cipher1.txt x cipher2.txt x
1|Salted__\87\0B-\89eP\0B\00\A0K\C7>\88\9FK\DA\00\&o\Ed\00\B0\E8=[ ]\EDz
```

Result from cipher2.txt:

```
cipher2.txt (~/Desktop/lab_4/task6/TASK 6.1/TWO DIFFRENT IVs) - gedit
Open + Save
cipher1.txt x cipher2.txt x
1 Salted__«n 0B7ygJY}UÚÊÉBS8Eke²êÈcá
Plain Text Tab Width: 8 Ln 1, Col 1 INS
```

b) same IVs:

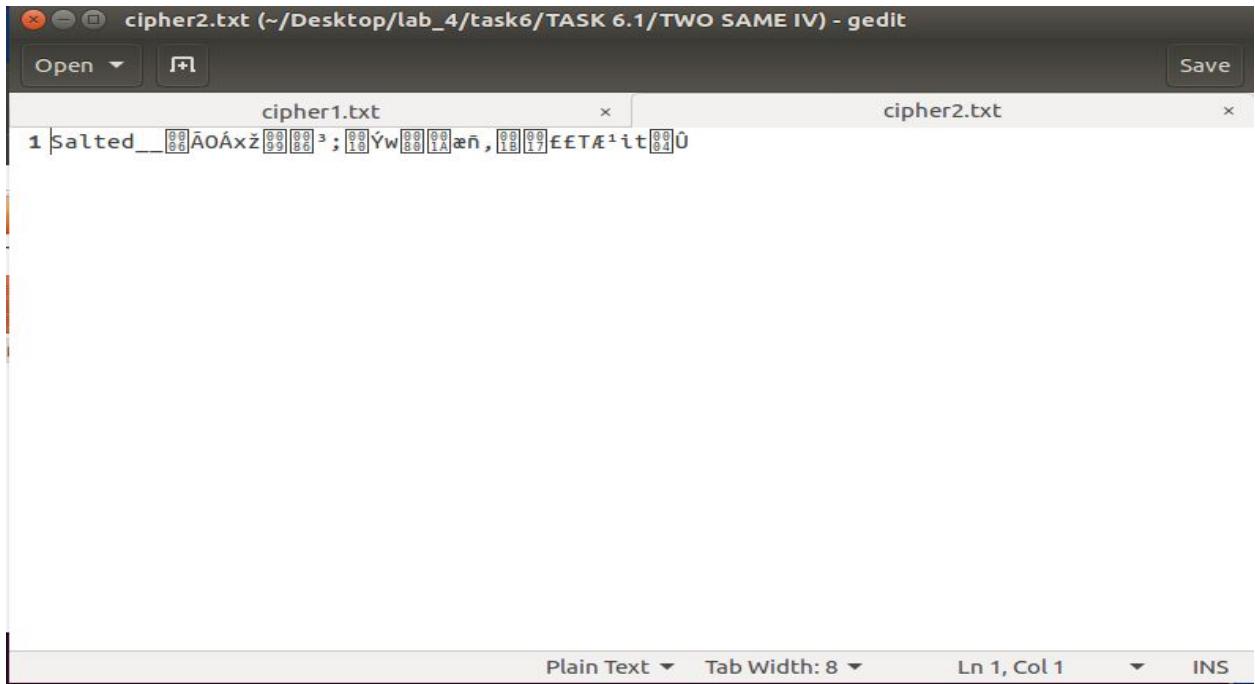
COMMAND:

```
[06/26/18]seed@VM:~/.../TWO SAME IV$ openssl enc -aes-128-of  
p -e -in plain.txt -out cipher1.txt \-k 00112233445566778889  
aabbcdddeeff \-iv 0102030405060708  
[06/26/18]seed@VM:~/.../TWO SAME IV$ openssl enc -aes-128-of  
p -e -in plain.txt -out cipher2.txt \-k 00112233445566778889  
aabbcdddeeff \-iv 0102030405060708  
[06/26/18]seed@VM:~/.../TWO SAME IV$ █
```

Result cipher1.txt:



Result cipher2.txt:



Observation: i think the formula that use for OFB mode is $V_i = C_i \oplus P_1$ so as long as the IV different the encryption will be changing. On the other hand those on the same IVs the block combine be the same context but different block cipher bits changing.

Task 6.2. One may argue that if the plaintext does not repeat, using the same IV is safe.

Solution: i think he/she decrypt other encrypted messages if the IV is always the same IVs as the formular:

OFB mode => $V_i = C_i \oplus P_1$

$C_1 = V_1 \oplus P_1$ then finding the IV $\Rightarrow V_1 = C_1 \oplus P_1$

$C_2 = V_1 \oplus P_2$ so we have $P_2 = V_1 \oplus C_2$

$\Rightarrow P_2 = C_1 \oplus P_1$ (1)

By following the substitution formula (1):
plain

54 68 69 73 20 69 73 20 61 20 6b 6e 6f 77 6e 20 6d 65 73 73 61 67 65 21

cipher 1

a4 69 b1 c5 02 c1 ca b9 66 96 5e 50 42 54 38 e1 bb 1b 5f 90 37 a4 c1 59

solution p1 xor c1

f0 1 d8 b6 22 a8 b9 99 7 b6 35 3e 2d 23 56 c1 d6 7e 2c e3 56 c3 a4 78

cipher 2

bf 73 bc d3 50 92 99 d5 66 c3 5b 5d 45 03 37 e1 bb 17 5f 90 3f af c1 59

p1 xor c1 xor c2

Message of p2 is: "order: Launch a missile! "

CFB MODE => Ci = Ek(Vi-1) ⊕ Pi

C1 = Ek (Vo) ⊕ P1

V0 = P1 ⊕ C1

As same IV

C2 = EK (V0) ⊕ P2

P2 = C2 ⊕ V0

P2 = P1 ⊕ C1 ⊕ C2

So the message with the same IV will have the same context of OFB

Message of p2 is: "order: Launch a missile! "

TASK 6.3:

By guessing IV assume that $C_i = V_i \oplus P_i$

And as the result we can see clearly that the message encryption

```
Encryption method: 128-bit AES with CBC mode.

Key (in hex): 00112233445566778899aabbccddeeff (known only to Bob)
Ciphertext (C1): bef65565572ccee2a9f9553154ed9498 (known to both)
IV used on P1 (known to both)
    (in ascii): 1234567890123456
    (in hex)   : 31323334353637383930313233343536
Next IV (known to both)
    (in ascii): 1234567890123457
    (in hex)   : 31323334353637383930313233343537
```

The iv for p1 and p2 is increasing by one from 36 to 37 so in the case the IV can be predictable.

$$C1 = IV1 \oplus P1 \oplus KEY$$

$$2) IV1 = IV(\text{unknown}) \oplus KEY \text{ assume that iv unknown first block}$$

Substitution 2 to 1

$$\Rightarrow C1 = IV(\text{unknown}) \oplus KEY \oplus P1 \oplus KEY$$

$$\Rightarrow C1 = IV(\text{unknown}) \oplus P1$$

Try to give to Bod the plaintext.txt with content “yes” in the first block then the equation will be

$$1) C2 = IV2 \oplus P2$$

Now we have (USING COMPARISON)

$$C2 = IV2 \oplus P2 = IV(\text{unknown}) \oplus P1 = C1$$

If $C2 = C1$ then context is “ YES ”

If $C_2 \neq C_1$ Then context is “ NO ”