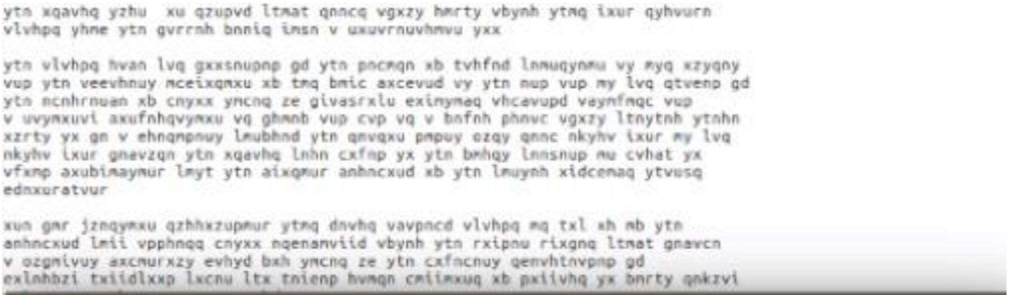
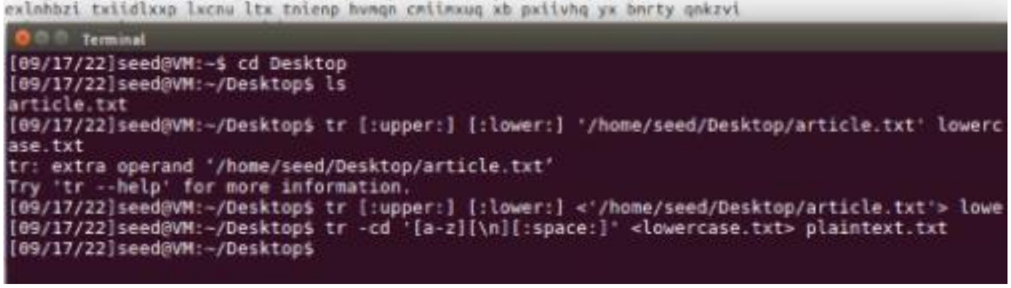
Group 8: Kevin Vo, Mason Godfrey, Thanh Le

CS 5500 – Lab 2: Symmetric Cryptography

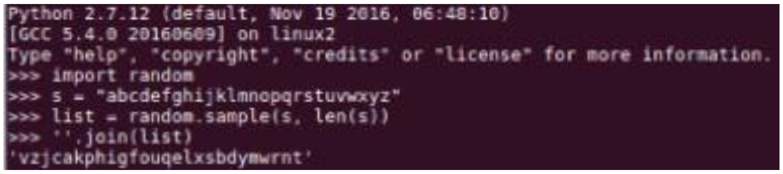
Task 1



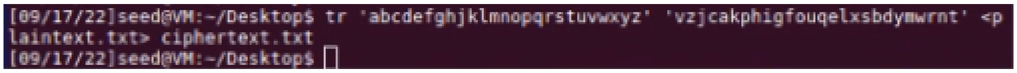
```
ytn xqavhq yzhu  xu qzupvd ltmat qnncq vgxzy hmrty vbynh ytmq ixur qyhvurn
vlvhpq yhme ytn gvrrnh bnniq imsn v uxuvrnuvhmvu yxx

ytn vlvhpq hvan lvq gxxsnupnp gd ytn pncmqn xb tvhfnd lnmuqynmu vy myq xzyqny
vup ytn veevhnuy mceixqmxu xb tmq bmic axcevud vy ytn nup vup my lvq qtvenp gd
ytn ncnhrnuan xb cnyxx ymcnq ze givasrxlu eximymaq vhcavupd vaymfmqc vup
v uvymxuvi axufnhqvymxu vq ghmnb vup cvp vq v bnfnh phnvc vgxzy ltnytnh ytnhn
xzrty yx gn v ehnqmpnuy lmubhnd ytn qnvqxu pmpuy ozqy qnnc nkyhv ixur my lvq
nkyhv ixur gnevzqn ytn xqavhq lnhn cxfnp yx ytn bmhqy lnnsnup mu cvhat yx
vfxmp axubimaymur lmyt ytn aixqmur anhncxud xb ytn lmuynh xidcenaq ytvusq
ednxuratvur

xun gmr jznqymxu qzhhxzupmur ytmq dnvhq vavpncd vlvhpq mq txl xh mb ytn
anhncxud lmii vpphnqq cnyxx nqenanviid vbynh ytn rxipnu rixgnq ltmat gnavcn
v ozgmivuy axcmurxzy evhyd bxh ymcnq ze ytn cxfncnuy qenvhtnvpnp gd
exlnhbzi txiidlxxp lxcnu ltx tnienp hvmqn cmiimxuq xb pxiivhq yx bnrty qnkzvi
```



```
[09/17/22]seed@VM:~$ cd Desktop
[09/17/22]seed@VM:~/Desktop$ ls
article.txt
[09/17/22]seed@VM:~/Desktop$ tr [:upper:] [:lower:] '/home/seed/Desktop/article.txt' lowerc
ase.txt
tr: extra operand '/home/seed/Desktop/article.txt'
Try 'tr --help' for more information.
[09/17/22]seed@VM:~/Desktop$ tr [:upper:] [:lower:] <'/home/seed/Desktop/article.txt'> lowe
[09/17/22]seed@VM:~/Desktop$ tr -cd '[a-z][\n][:space:]' <lowercase.txt> plaintext.txt
[09/17/22]seed@VM:~/Desktop$
```



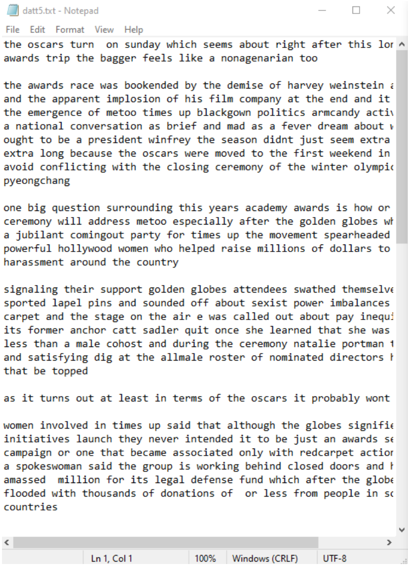
```
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import random
>>> s = "abcdefghijklmnopqrstuvwxyz"
>>> list = random.sample(s, len(s))
>>> ''.join(list)
'vzjcakphigfouqelxsbdymwrnt'
```



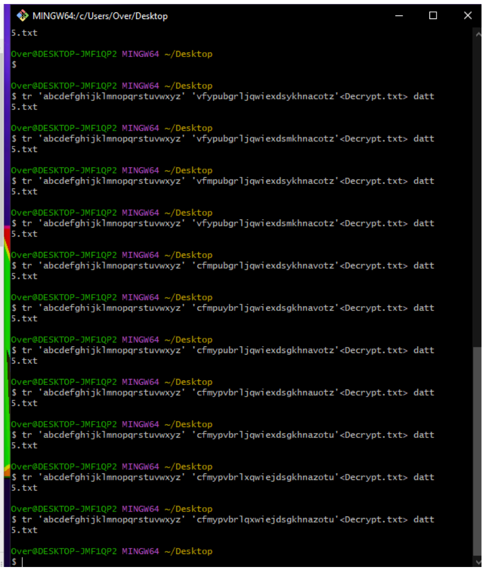
```
[09/17/22]seed@VM:~/Desktop$ tr 'abcdefghjklmnopqrstuvwxyz' 'vzjcakphigfouqelxsbdymwrnt' <p
laintext.txt> ciphertext.txt
[09/17/22]seed@VM:~/Desktop$ 
```

Your job is to use the frequency analysis to figure out the encryption key and the original plaintext.

The encryption key was ‘cfmypvbrlqxwiejdsgkhnazotu’. The following is the original plaintext.



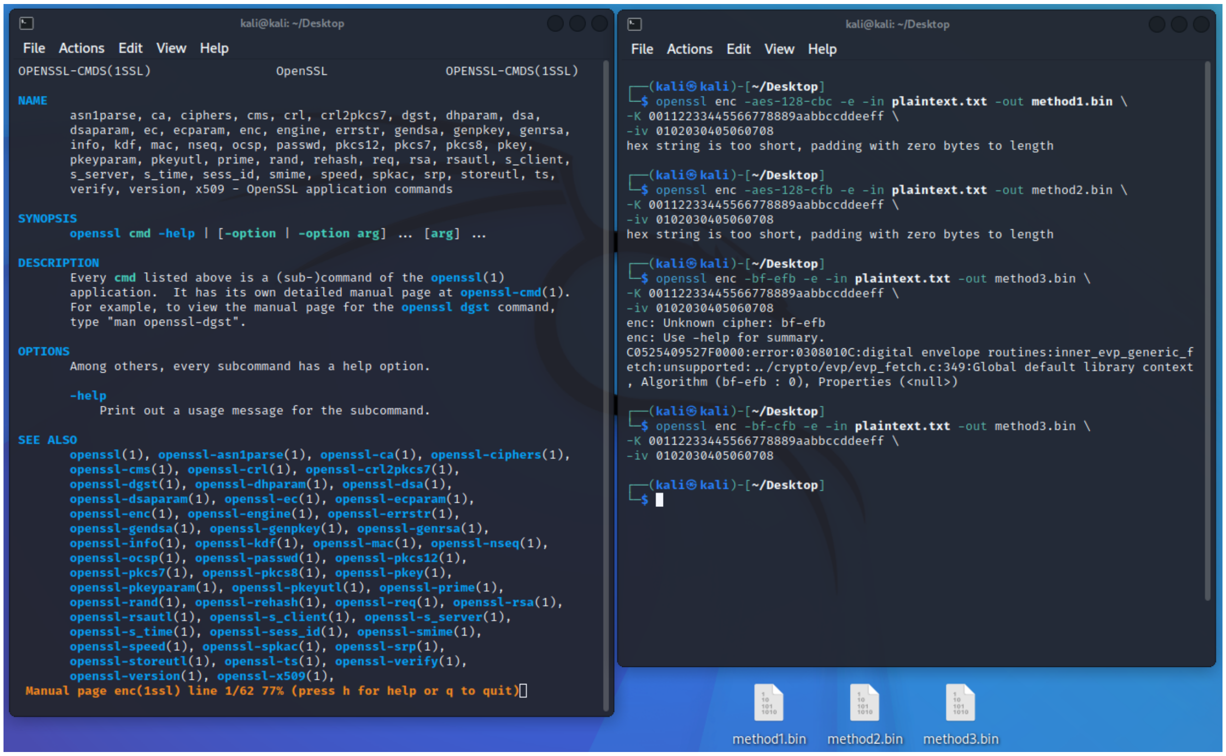
```
the oscars turn  on sunday which seems about right after this lor
awards trip the bagger feels like a nonagenarian too

the awards race was bookended by the demise of harvey weinstein a
and the apparent implosion of his film company at the end and it
the emergence of metoo times up blackgown politics armcandy activ
a national conversation as brief and mad as a fever dream about w
ought to be a president winfrey the season didnt just seem extra
extra long because the oscars were moved to the first weekend in
avoid conflicting with the closing ceremony of the winter olympic
pyeongchang

one big question surrounding this years academy awards is how or
ceremony will address metoo especially after the golden globes wh
a jubilant comingout party for times up the movement spearheaded
powerful hollywood women who helped raise millions of dollars to
harassment around the country

signaling their support golden globes attendees swathed themselve
sported lapel pins and sounded off about sexist power imbalances
carpet and the stage on the air e was called out about pay inequi
its former anchor catt sadler quit once she learned that she was
less than a male cohost and during the ceremony natalie portman t
and satisfying dig at the allmale roster of nominated directors h
that be topped

as it turns out at least in terms of the oscars it probably wont

women involved in times up said that although the globes signifie
initiatives launch they never intended it to be just an awards se
campaign or one that became associated only with redcarpet actior
a spokeswoman said the group is working behind closed doors and h
amassed  million for its legal defense fund which after the globe
flooded with thousands of donations of  or less from people in sc
countries
```



```
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'vfypubgrljqwiexdsykhnacotz'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'vfypubgrljqwiexdsmkhnacotz'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'vfmpubgrljqwiexdsykhnacotz'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'vfypubgrljqwiexdsmkhnacotz'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'cfmpubgrljqwiexdsykhnavotz'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'cfmpuybrljqwiexdsgkhnavotz'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'cfmypubrljqwiexdsgkhnavotz'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'cfmypvbrljqwiexdsgkhnauotz'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'cfmypvbrljqwiexdsgkhnazotu'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'cfmypvbrlxqwiejdsgkhnazotu'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ tr 'abcdefghijklmnopqrstuvwxyz' 'cfmypvbrlqxwiejdsgkhnazotu'<Decrypt.txt> datt
5.txt

Over@DESKTOP-JMF1QP2 MINGW64 ~/Desktop
$ 
```

Task 2

Please replace the ciphertype with a specific cipher type, such as -aes-128-cbc, -bf-cbc, -aes-128-cfb, etc. In this task, you should try at least 3 different ciphers.

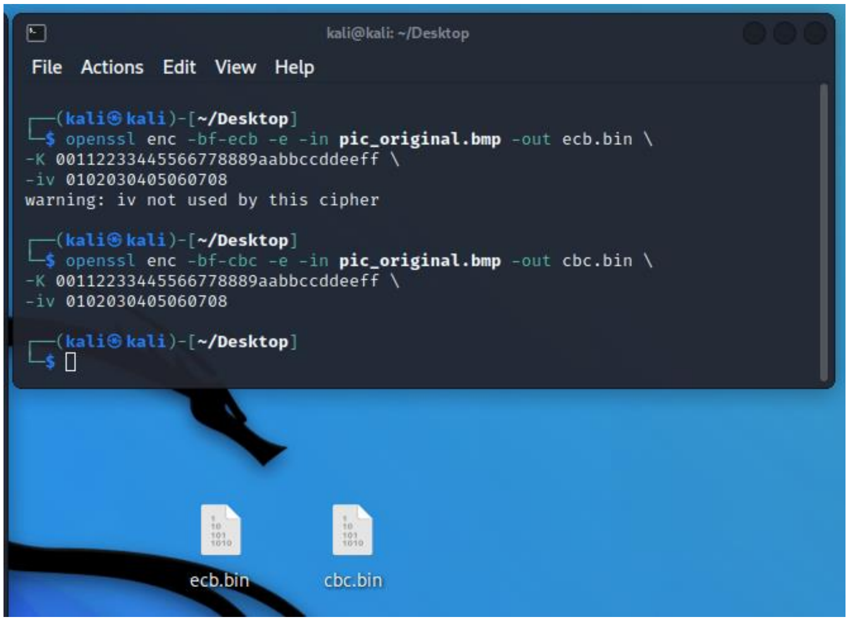


```
OPENSSL-CMDS(1SSL)                OpenSSL               OPENSSL-CMDS(1SSL)

NAME
       asn1parse, ca, ciphers, cms, crl, crl2pkcs7, dgst, dhparam, dsa,
       dsaparam, ec, ecparam, enc, engine, errstr, gendsa, genpkey, genrsa,
       info, kdf, mac, nseq, ocsp, passwd, pkcs12, pkcs7, pkcs8, pkey,
       pkeyparam, pkeyutl, prime, rand, rehash, req, rsa, rsautl, s_client,
       s_server, s_time, sess_id, smime, speed, spkac, srp, storeutl, ts,
       verify, version, x509 - OpenSSL application commands

SYNOPSIS
       openssl cmd -help | [-option | -option arg] ... [arg] ...

DESCRIPTION
       Every cmd listed above is a (sub-)command of the openssl(1)
       application.  It has its own detailed manual page at openssl-cmd(1).
       For example, to view the manual page for the openssl dgst command,
       type "man openssl-dgst".

OPTIONS
       Among others, every subcommand has a help option.

       -help
           Print out a usage message for the subcommand.

SEE ALSO
       openssl(1), openssl-asn1parse(1), openssl-ca(1), openssl-ciphers(1),
       openssl-cms(1), openssl-crl(1), openssl-crl2pkcs7(1),
       openssl-dgst(1), openssl-dhparam(1), openssl-dsa(1),
       openssl-dsaparam(1), openssl-ec(1), openssl-ecparam(1),
       openssl-enc(1), openssl-engine(1), openssl-errstr(1),
       openssl-gendsa(1), openssl-genpkey(1), openssl-genrsa(1),
       openssl-info(1), openssl-kdf(1), openssl-mac(1), openssl-nseq(1),
       openssl-ocsp(1), openssl-passwd(1), openssl-pkcs12(1),
       openssl-pkcs7(1), openssl-pkcs8(1), openssl-pkey(1),
       openssl-pkeyparam(1), openssl-pkeyutl(1), openssl-prime(1),
       openssl-rand(1), openssl-rehash(1), openssl-req(1), openssl-rsa(1),
       openssl-rsautl(1), openssl-s_client(1), openssl-s_server(1),
       openssl-s_time(1), openssl-sess_id(1), openssl-smime(1),
       openssl-speed(1), openssl-spkac(1), openssl-srp(1),
       openssl-storeutl(1), openssl-ts(1), openssl-verify(1),
       openssl-version(1), openssl-x509(1),
 Manual page enc(1ssl) line 1/62 77% (press h for help or q to quit)
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cbc -e -in plaintext.txt -out method1.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cfb -e -in plaintext.txt -out method2.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -bf-efb -e -in plaintext.txt -out method3.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
enc: Unknown cipher: bf-efb
enc: Use -help for summary.
C0525409527F0000:error:0308010C:digital envelope routines:inner_evp_generic_f
etch:unsupported:../crypto/evp/evp_fetch.c:349:Global default library context
, Algorithm (bf-efb : 0), Properties (<null>)

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -bf-cfb -e -in plaintext.txt -out method3.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

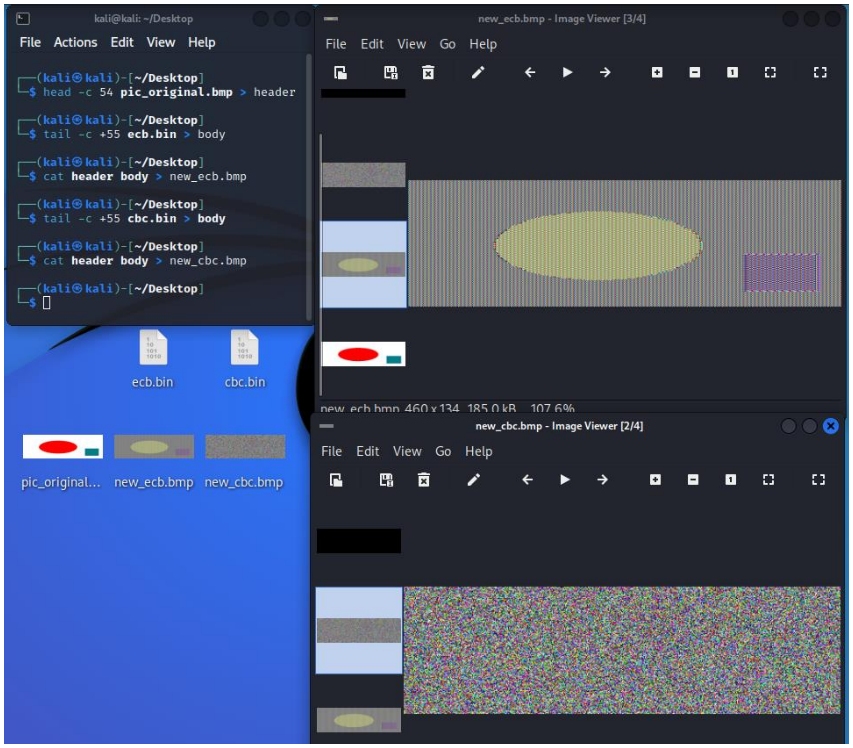
Task 3

Please encrypt the file using the ECB (Electronic Code Book) and CBC (Cipher Block Chaining) modes.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -bf-ecb -e -in pic_original.bmp -out ecb.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
warning: iv not used by this cipher

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -bf-cbc -e -in pic_original.bmp -out cbc.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708

┌──(kali㉿kali)-[~/Desktop]
└─$ ▯
```

We will replace the header of the encrypted picture with that of the original picture. Can you derive any useful information about the original picture from the encrypted picture? Please explain your observations.



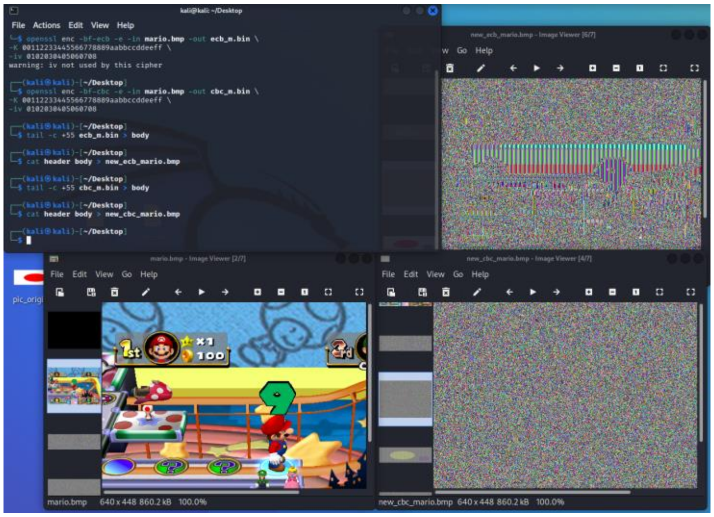
```
┌──(kali㉿kali)-[~/Desktop]
└─$ head -c 54 pic_original.bmp > header

┌──(kali㉿kali)-[~/Desktop]
└─$ tail -c +55 ecb.bin > body

┌──(kali㉿kali)-[~/Desktop]
└─$ cat header body > new_ecb.bmp

┌──(kali㉿kali)-[~/Desktop]
└─$ tail -c +55 cbc.bin > body

┌──(kali㉿kali)-[~/Desktop]
└─$ cat header body > new_cbc.bmp

┌──(kali㉿kali)-[~/Desktop]
└─$ ▯
```

kali@kali: ~/Desktop
File Actions Edit View Help
new_ecb.bmp - Image Viewer [3/4]
File Edit View Go Help
new_ecb.bmp 460 x 134 185.0 kB 107.6%
new_cbc.bmp - Image Viewer [2/4]
File Edit View Go Help
ecb.bin
cbc.bin
pic_original...
new_ecb.bmp
new_cbc.bmp

ECB mode: We can still derive some resemblance about the original picture from the encrypted picture. The shape and location of the oval and rectangle objects are recognized even though the colors of them are distorted because of pattern noise.

CBC mode: We cannot derive any information about the original picture from the encrypted picture. The encrypted picture just appears with full of random noise with absolutely no traces of the original picture.

Repeating the tasks:

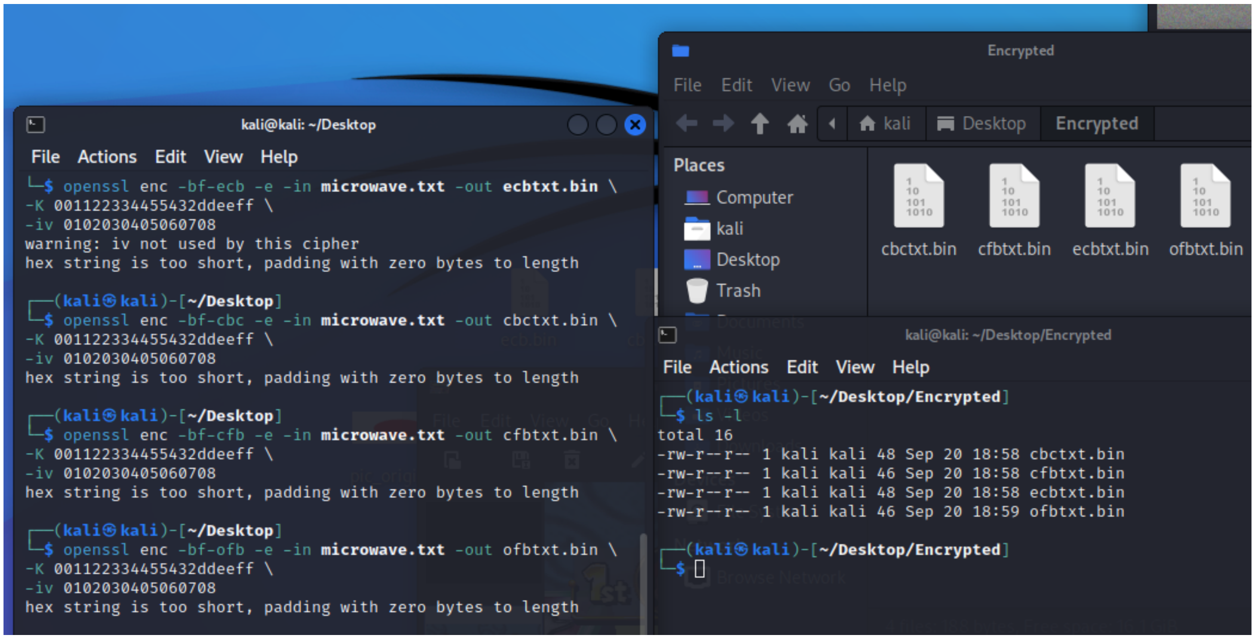


```
└─$ openssl enc -bf-ecb -e -in mario.bmp -out ecb_m.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
warning: iv not used by this cipher

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -bf-cbc -e -in mario.bmp -out cbc_m.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708

┌──(kali㉿kali)-[~/Desktop]
└─$ tail -c +55 ecb_m.bin > body

┌──(kali㉿kali)-[~/Desktop]
└─$ cat header body > new_ecb_mario.bmp

┌──(kali㉿kali)-[~/Desktop]
└─$ tail -c +55 cbc_m.bin > body

┌──(kali㉿kali)-[~/Desktop]
└─$ cat header body > new_cbc_mario.bmp

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

kali@kali: ~/Desktop
File Actions Edit View Help
new_ecb_mario.bmp - Image Viewer [6/7]
mario.bmp - Image Viewer [2/7]
File Edit View Go Help
pic_origi
mario.bmp 640 x 448 860.2 kB 100.0%
new_cbc_mario.bmp - Image Viewer [4/7]
new_cbc_mario.bmp 640 x 448 860.2 kB 100.0%

For the Mario screenshot, while CBC file still seems like a random image, a few locations in the ECB encrypted file bleed through; the number 9 can be seen, and areas where colors don't change much in the original image are similar in the new image.

Task 4



```
└─$ openssl enc -bf-ecb -e -in microwave.txt -out ecbtxt.bin \
-K 00112233445543 2ddeeff \
-iv 0102030405060708
warning: iv not used by this cipher
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -bf-cbc -e -in microwave.txt -out cbctxt.bin \
-K 001122334455432ddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -bf-cfb -e -in microwave.txt -out cfbtxt.bin \
-K 001122334455432ddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -bf-ofb -e -in microwave.txt -out ofbtxt.bin \
-K 001122334455432ddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length
```

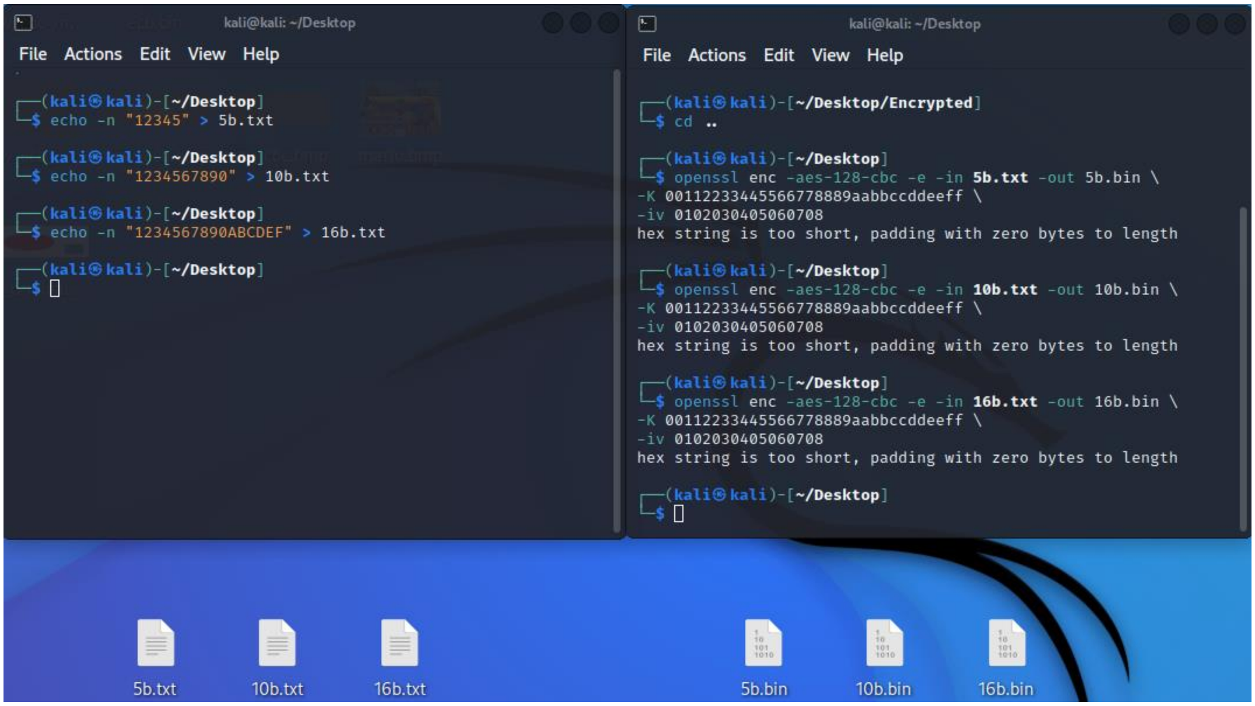
```
┌──(kali㉿kali)-[~/Desktop/Encrypted]
└─$ ls -l
total 16
-rw-r--r-- 1 kali kali 48 Sep 20 18:58 cbctxt.bin
-rw-r--r-- 1 kali kali 46 Sep 20 18:58 cfbtxt.bin
-rw-r--r-- 1 kali kali 48 Sep 20 18:58 ecbtxt.bin
-rw-r--r-- 1 kali kali 46 Sep 20 18:59 ofbtxt.bin

┌──(kali㉿kali)-[~/Desktop/Encrypted]
└─$ ▯
```

Please report which modes have paddings and which ones do not. For those that do not need paddings, please explain why.

CBC and ECB need padding while CFB and OFB do not. This can be determined by looking at their lengths. CFB and OFB do not need padding as both allow for input text of any length for each block so long as the output block is the same size as the input block. Then, the block size can be any size including 1. The files, then, don't need to be padded.

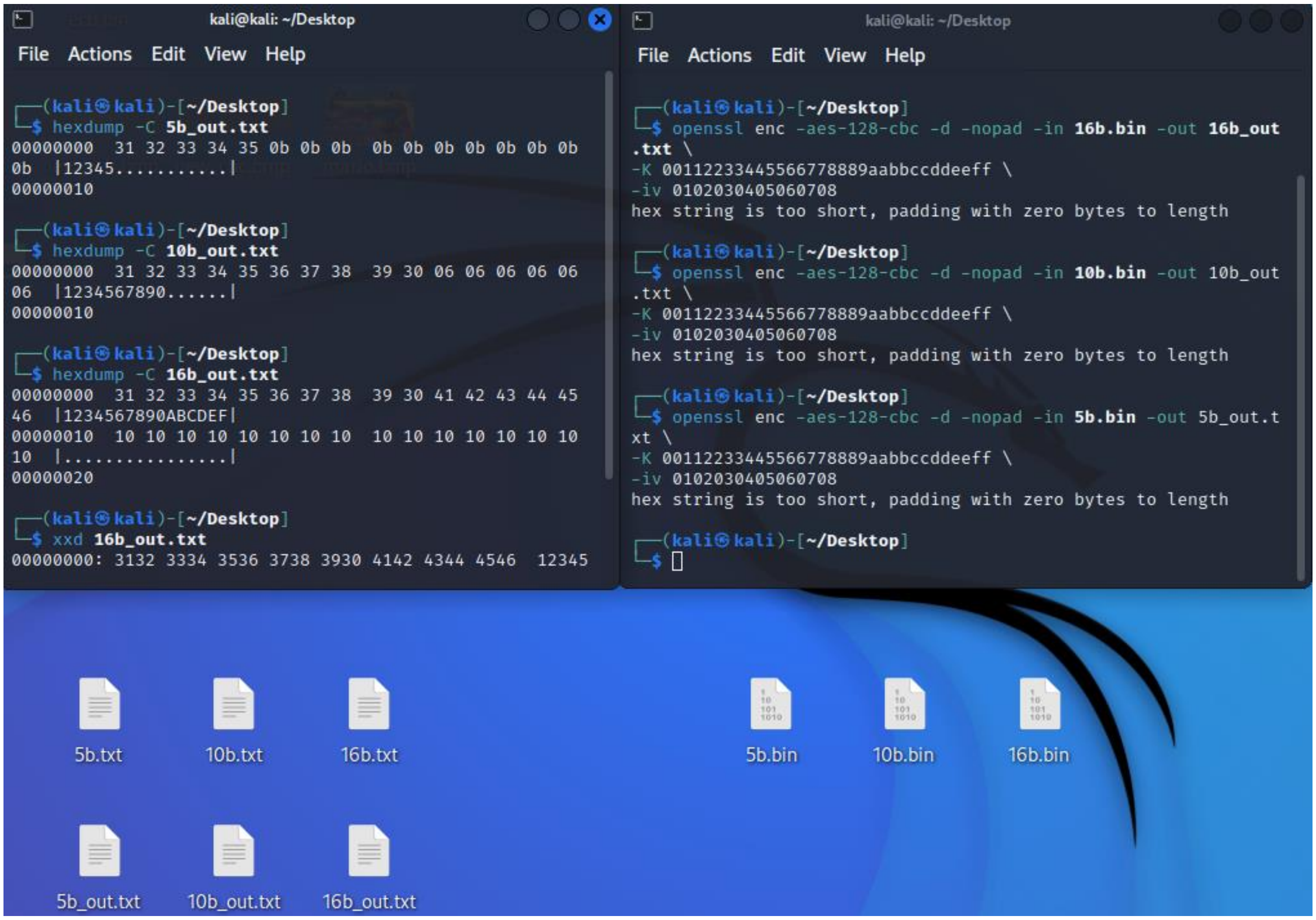
In other words, like the instructions for this task has stated, padding is applied if the plaintexts are not part of the block size. Which are multiples of 16 bytes. For example, if we have a plaintext of less than 16 bytes, it will pad enough characters until it hits 16 bytes to reach the block size. However, if we have a plaintext of 16 bytes and below 32 bytes, we'd have a file that'd padded until it hits 32 bytes. From our observation, this matches what the instructions had said about how padding operates. Based on our images with the padding at the end of each plaintext, the paddings fit accordingly.

Below is image of our commands for CBC, CFB, ECB, and OFB encrypting of 5-, 10-, and 16-bytes plaintexts. The tables show the sizes that we observe of each encrypted and the size of its decrypted files. With the "-nopad" option we were able to see which encryption uses padding. Without padding the contents of the decrypted files should only contain numbers from [0,9] just like its original plaintext.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n "12345" > 5b.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n "1234567890" > 10b.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n "1234567890ABCDEF" > 16b.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ ▯
```

```
┌──(kali㉿kali)-[~/Desktop/Encrypted]
└─$ cd ..

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cbc -e -in 5b.txt -out 5b.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cbc -e -in 10b.txt -out 10b.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cbc -e -in 16b.txt -out 16b.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ ▯
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ hexdump -C 5b_out.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b
0b  |12345...........|
00000010

┌──(kali㉿kali)-[~/Desktop]
└─$ hexdump -C 10b_out.txt
00000000  31 32 33 34 35 36 37 38  39 30 06 06 06 06 06
06  |1234567890......|
00000010

┌──(kali㉿kali)-[~/Desktop]
└─$ hexdump -C 16b_out.txt
00000000  31 32 33 34 35 36 37 38  39 30 41 42 43 44 45
46  |1234567890ABCDEF|
00000010  10 10 10 10 10 10 10 10  10 10 10 10 10 10 10
10  |................|
00000020

┌──(kali㉿kali)-[~/Desktop]
└─$ xxd 16b_out.txt
00000000: 3132 3334 3536 3738 3930 4142 4344 4546  12345
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cbc -d -nopad -in 16b.bin -out 16b_out
.txt \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cbc -d -nopad -in 10b.bin -out 10b_out
.txt \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cbc -d -nopad -in 5b.bin -out 5b_out.t
xt \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ ▯
```

<u>The following entry below for remainder of Task 4 was from a different computer. With a different naming convention. It covers what was said above with some extra details.</u>

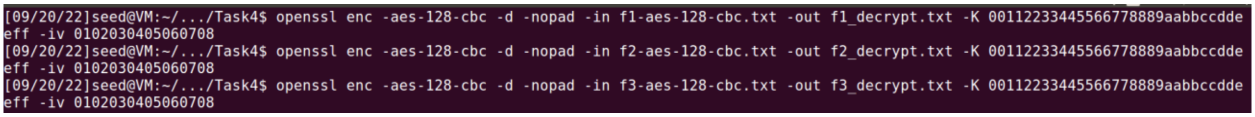
Uses Padding: CBC, ECB

Does NOT Use Padding: CFB, OFB

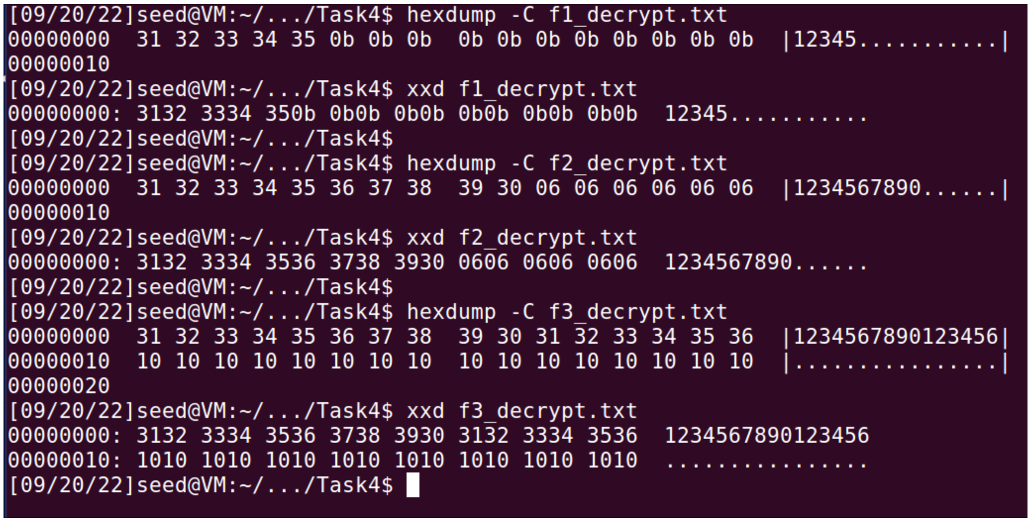
From observing the results above, it seems that the CBC and ECB encryptions uses padding. On the other hand, encryptions CFB and OFB do NOT use padding. The reason behind CFB and OFB NOT using padding is because it contains more block sizes instead of ones that are multiple of 16.

We could tell based on observing the decrypted contents of each 5-, 10-, and 16-bytes plaintext along with the increase in size of its encrypted files. Finally, another indicator we found that an encryption that uses padding is in-use is from the display of the "hexdump" command as there were no fillers with the "…" displayed for the non-padding encryptions.

Using CBC (YES padding):

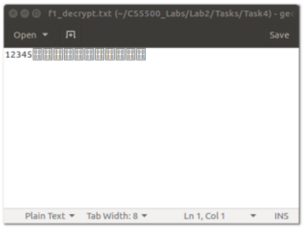


```
[09/20/22]seed@VM:~/.../Task4$ openssl enc -aes-128-cbc -d -nopad -in f1-aes-128-cbc.txt -out f1_decrypt.txt -K 00112233445566778889aabbccdde
eff -iv 0102030405060708
[09/20/22]seed@VM:~/.../Task4$ openssl enc -aes-128-cbc -d -nopad -in f2-aes-128-cbc.txt -out f2_decrypt.txt -K 00112233445566778889aabbccdde
eff -iv 0102030405060708
[09/20/22]seed@VM:~/.../Task4$ openssl enc -aes-128-cbc -d -nopad -in f3-aes-128-cbc.txt -out f3_decrypt.txt -K 00112233445566778889aabbccdde
eff -iv 0102030405060708
```

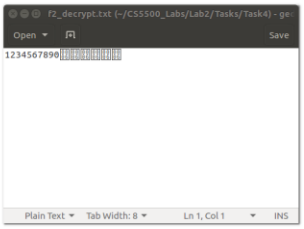


```
[09/20/22]seed@VM:~/.../Task4$ hexdump -C f1_decrypt.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b 0b  |12345...........|
00000010
[09/20/22]seed@VM:~/.../Task4$ xxd f1_decrypt.txt
00000000: 3132 3334 350b 0b0b 0b0b 0b0b 0b0b 0b0b  12345...........
[09/20/22]seed@VM:~/.../Task4$
[09/20/22]seed@VM:~/.../Task4$ hexdump -C f2_decrypt.txt
00000000  31 32 33 34 35 36 37 38  39 30 06 06 06 06 06 06  |1234567890......|
00000010
[09/20/22]seed@VM:~/.../Task4$ xxd f2_decrypt.txt
00000000: 3132 3334 3536 3738 3930 0606 0606 0606  1234567890......
[09/20/22]seed@VM:~/.../Task4$
[09/20/22]seed@VM:~/.../Task4$ hexdump -C f3_decrypt.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010  10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10  |................|
00000020
[09/20/22]seed@VM:~/.../Task4$ xxd f3_decrypt.txt
00000000: 3132 3334 3536 3738 3930 3132 3334 3536  1234567890123456
00000010: 1010 1010 1010 1010 1010 1010 1010 1010  ................
[09/20/22]seed@VM:~/.../Task4$
```

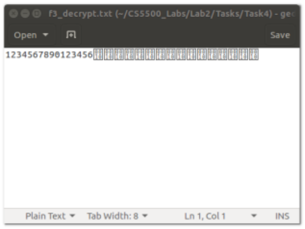
The image below shows that CBC does use padding due to there being unrecognizable character of the decrypted files and its plain text at the bottom. From right to left are the 5-bytes, 10-bytes, and 16-bytes respectively.



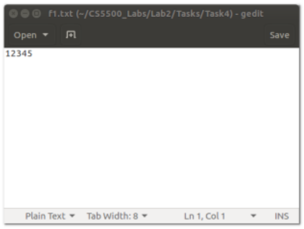
```
12345▯▯▯▯▯▯▯▯▯▯▯
```



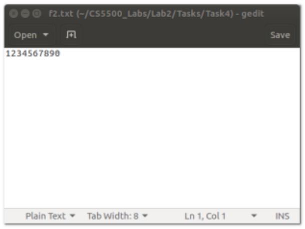
```
1234567890▯▯▯▯▯▯
```



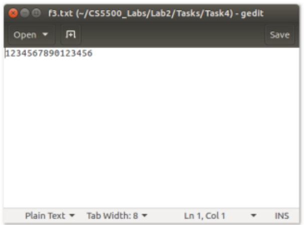
```
1234567890123456▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
```



```
12345
```



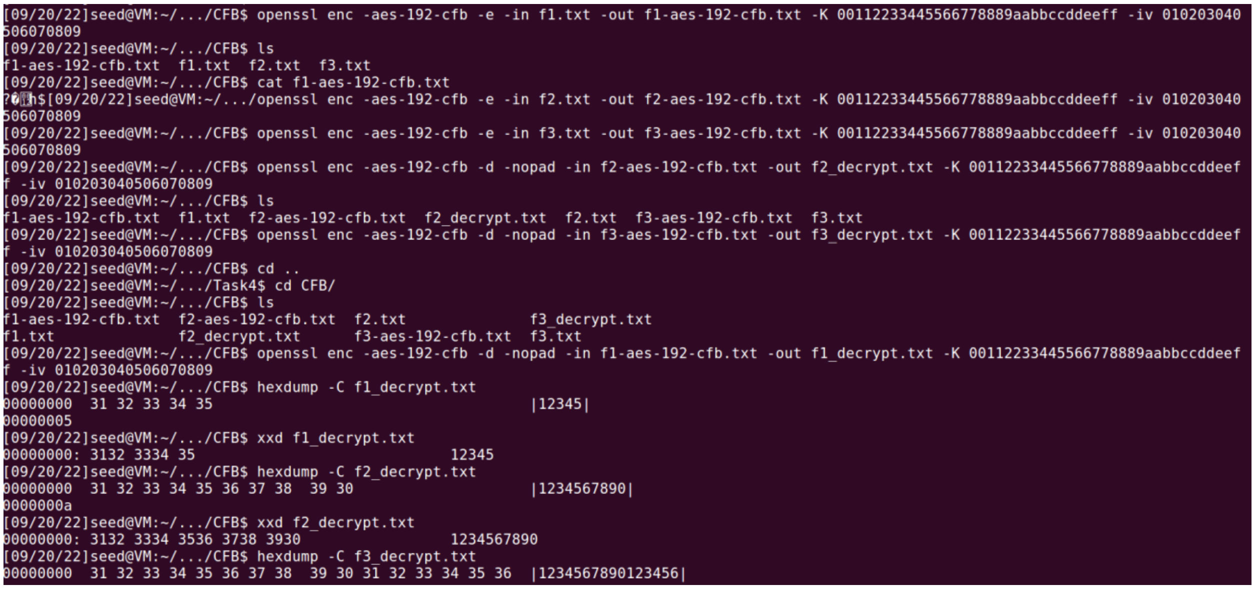
```
1234567890
```



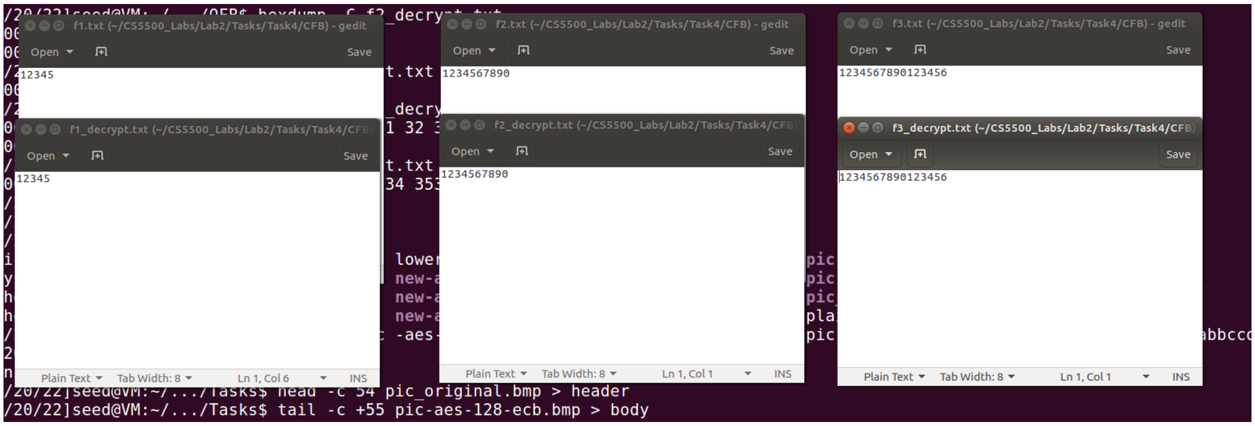
```
1234567890123456
```

Using CBC (-aes-128-cbc)		
Original Filename	Original Size (Bytes)	Padded Size (Bytes)
f1.txt	5	16
f2.txt	10	16
f3.txt	16	32
CBC is Padded		

Using CFB (NO padded):



```
[09/20/22]seed@VM:~/.../CFB$ openssl enc -aes-192-cfb -e -in f1.txt -out f1-aes-192-cfb.txt -K 00112233445566778889aabbccddeeff -iv 010203040
506070809
[09/20/22]seed@VM:~/.../CFB$ ls
f1-aes-192-cfb.txt  f1.txt  f2.txt  f3.txt
[09/20/22]seed@VM:~/.../CFB$ cat f1-aes-192-cfb.txt
?�▯h$[09/20/22]seed@VM:~/.../openssl enc -aes-192-cfb -e -in f2.txt -out f2-aes-192-cfb.txt -K 00112233445566778889aabbccddeeff -iv 010203040
506070809
[09/20/22]seed@VM:~/.../CFB$ openssl enc -aes-192-cfb -e -in f3.txt -out f3-aes-192-cfb.txt -K 00112233445566778889aabbccddeeff -iv 010203040
506070809
[09/20/22]seed@VM:~/.../CFB$ openssl enc -aes-192-cfb -d -nopad -in f2-aes-192-cfb.txt -out f2_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 0102030405060708O9
[09/20/22]seed@VM:~/.../CFB$ ls
f1-aes-192-cfb.txt  f1.txt  f2-aes-192-cfb.txt  f2_decrypt.txt  f2.txt  f3-aes-192-cfb.txt  f3.txt
[09/20/22]seed@VM:~/.../CFB$ openssl enc -aes-192-cfb -d -nopad -in f3-aes-192-cfb.txt -out f3_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 010203040506070809
[09/20/22]seed@VM:~/.../CFB$ cd ..
[09/20/22]seed@VM:~/.../Task4$ cd CFB/
[09/20/22]seed@VM:~/.../CFB$ ls
f1-aes-192-cfb.txt  f2-aes-192-cfb.txt  f2.txt              f3_decrypt.txt
f1.txt              f2_decrypt.txt      f3-aes-192-cfb.txt  f3.txt
[09/20/22]seed@VM:~/.../CFB$ openssl enc -aes-192-cfb -d -nopad -in f1-aes-192-cfb.txt -out f1_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 010203040506070809
[09/20/22]seed@VM:~/.../CFB$ hexdump -C f1_decrypt.txt
00000000  31 32 33 34 35                                    |12345|
00000005
[09/20/22]seed@VM:~/.../CFB$ xxd f1_decrypt.txt
00000000: 3132 3334 35                             12345
[09/20/22]seed@VM:~/.../CFB$ hexdump -C f2_decrypt.txt
00000000  31 32 33 34 35 36 37 38  39 30                    |1234567890|
0000000a
[09/20/22]seed@VM:~/.../CFB$ xxd f2_decrypt.txt
00000000: 3132 3334 3536 3738 3930                 1234567890
[09/20/22]seed@VM:~/.../CFB$ hexdump -C f3_decrypt.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
```

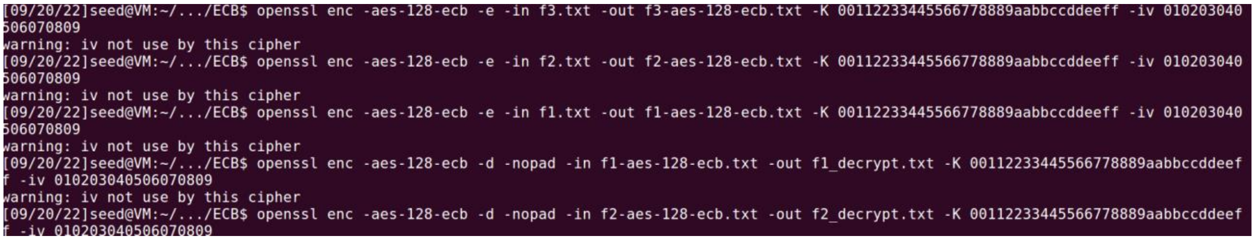


```
f1.txt (~/CS5500_Labs/Lab2/Tasks/Task4/CFB) - gedit
12345

f1_decrypt.txt (~/CS5500_Labs/Lab2/Tasks/Task4/CFB)
12345

f2.txt (~/CS5500_Labs/Lab2/Tasks/Task4/CFB) - gedit
1234567890

f2_decrypt.txt (~/CS5500_Labs/Lab2/Tasks/Task4/CFB)
1234567890

f3.txt (~/CS5500_Labs/Lab2/Tasks/Task4/CFB) - gedit
1234567890123456

f3_decrypt.txt (~/CS5500_Labs/Lab2/Tasks/Task4/CFB)
1234567890123456
```

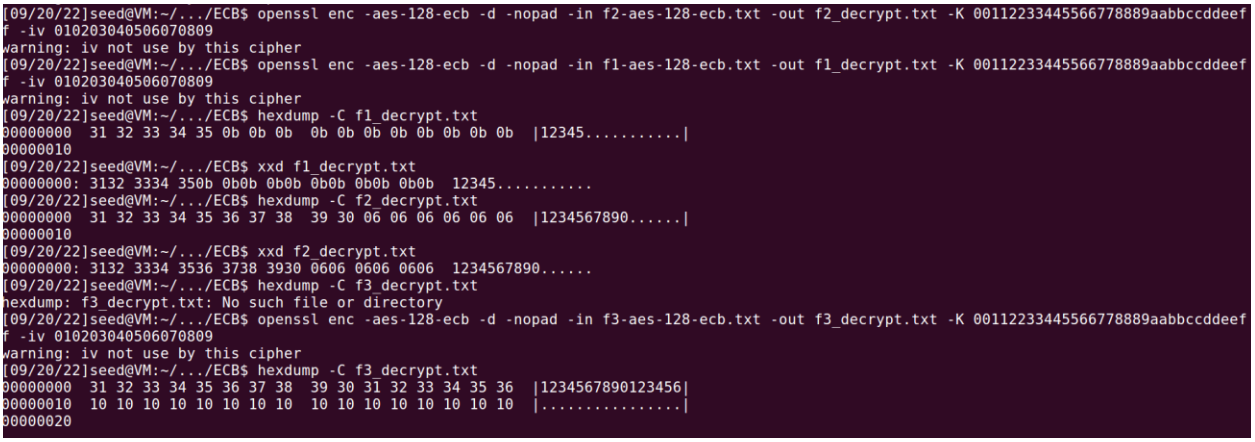
```
/20/22]seed@VM:~/.../Tasks$ head -c 54 pic_original.bmp > header
/20/22]seed@VM:~/.../Tasks$ tail -c +55 pic-aes-128-ecb.bmp > body
```

Using CFB (-aes-192-cfb)		
Original Filename	Original Size (Bytes)	Padded Size (Bytes)
f1.txt	5	5
f2.txt	10	10
f3.txt	16	16
CFB is NOT Padded		

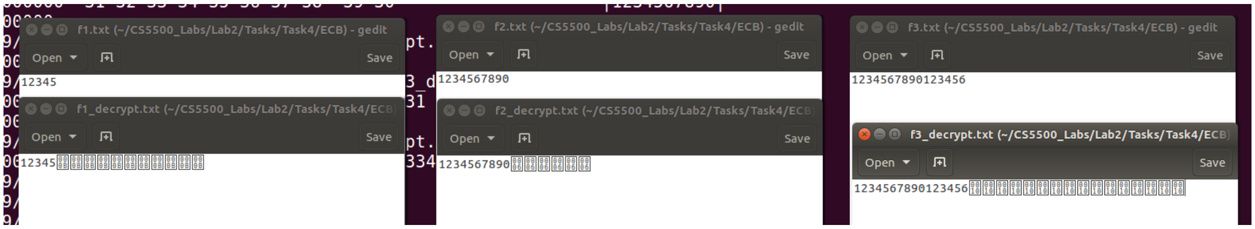
Using ECB (YES padding):



```
[09/20/22]seed@VM:~/.../ECB$ openssl enc -aes-128-ecb -e -in f3.txt -out f3-aes-128-ecb.txt -K 00112233445566778889aabbccddeeff -iv 010203040
506070809
warning: iv not use by this cipher
[09/20/22]seed@VM:~/.../ECB$ openssl enc -aes-128-ecb -e -in f2.txt -out f2-aes-128-ecb.txt -K 00112233445566778889aabbccddeeff -iv 010203040
506070809
warning: iv not use by this cipher
[09/20/22]seed@VM:~/.../ECB$ openssl enc -aes-128-ecb -e -in f1.txt -out f1-aes-128-ecb.txt -K 00112233445566778889aabbccddeeff -iv 010203040
506070809
warning: iv not use by this cipher
[09/20/22]seed@VM:~/.../ECB$ openssl enc -aes-128-ecb -d -nopad -in f1-aes-128-ecb.txt -out f1_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 010203040506070809
warning: iv not use by this cipher
[09/20/22]seed@VM:~/.../ECB$ openssl enc -aes-128-ecb -d -nopad -in f2-aes-128-ecb.txt -out f2_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 010203040506070809
```



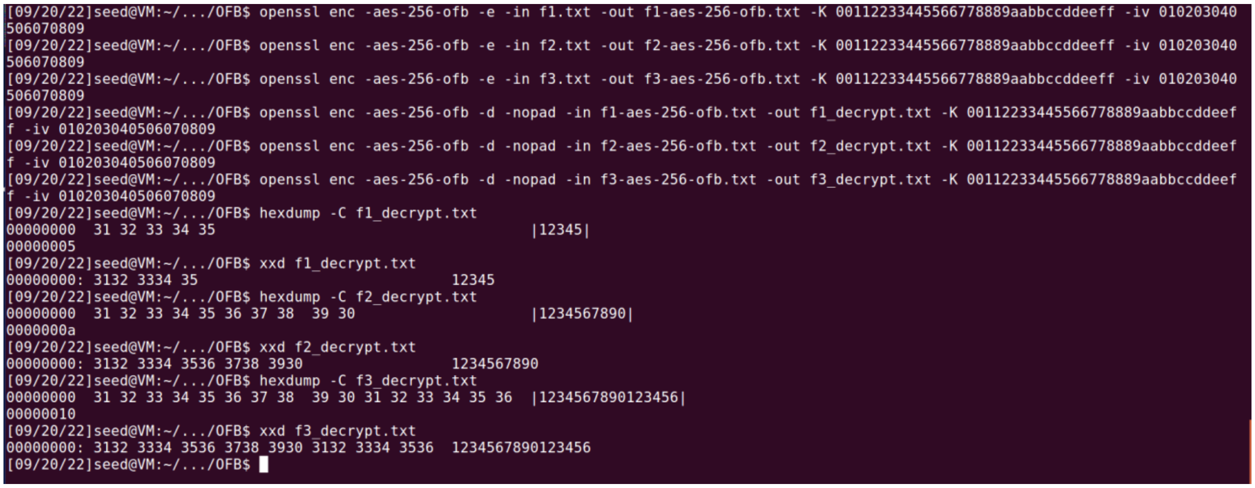
```
[09/20/22]seed@VM:~/.../ECB$ openssl enc -aes-128-ecb -d -nopad -in f2-aes-128-ecb.txt -out f2_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 0102030405060708‌09
warning: iv not use by this cipher
[09/20/22]seed@VM:~/.../ECB$ openssl enc -aes-128-ecb -d -nopad -in f1-aes-128-ecb.txt -out f1_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 010203040506070809
warning: iv not use by this cipher
[09/20/22]seed@VM:~/.../ECB$ hexdump -C f1_decrypt.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b 0b  |12345...........|
00000010
[09/20/22]seed@VM:~/.../ECB$ xxd f1_decrypt.txt
00000000: 3132 3334 350b 0b0b 0b0b 0b0b 0b0b 0b0b  12345...........
[09/20/22]seed@VM:~/.../ECB$ hexdump -C f2_decrypt.txt
00000000  31 32 33 34 35 36 37 38  39 30 06 06 06 06 06 06  |1234567890......|
00000010
[09/20/22]seed@VM:~/.../ECB$ xxd f2_decrypt.txt
00000000: 3132 3334 3536 3738 3930 0606 0606 0606  1234567890......
[09/20/22]seed@VM:~/.../ECB$ hexdump -C f3_decrypt.txt
hexdump: f3_decrypt.txt: No such file or directory
[09/20/22]seed@VM:~/.../ECB$ openssl enc -aes-128-ecb -d -nopad -in f3-aes-128-ecb.txt -out f3_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 010203040506070809
warning: iv not use by this cipher
[09/20/22]seed@VM:~/.../ECB$ hexdump -C f3_decrypt.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010  10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10  |................|
00000020
```


f1.txt (~/CS5500_Labs/Lab2/Tasks/Task4/ECB) - gedit
Open
Save
f1_decrypt.txt (~/CS5500_Labs/Lab2/Tasks/Task4/ECB
Open
Save
f2.txt (~/CS5500_Labs/Lab2/Tasks/Task4/ECB) - gedit
Open
Save
f2_decrypt.txt (~/CS5500_Labs/Lab2/Tasks/Task4/ECB
Open
Save
f3.txt (~/CS5500_Labs/Lab2/Tasks/Task4/ECB) - gedit
Open
Save
f3_decrypt.txt (~/CS5500_Labs/Lab2/Tasks/Task4/ECB)
Open
Save

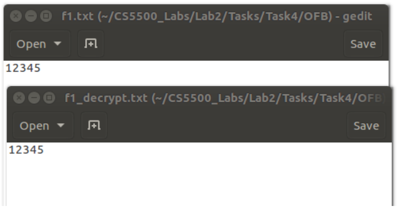
```
12345
12345
1234567890
1234567890
1234567890123456
1234567890123456
```

Using ECB (-aes-128-ecb)		
Original Filename	Original Size (Bytes)	Padded Size (Bytes)
f1.txt	5	16
f2.txt	10	16
f3.txt	16	32
ECB is Padded		

Using OFB (NO padding):

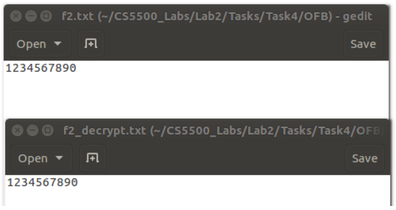


```
[09/20/22]seed@VM:~/.../OFB$ openssl enc -aes-256-ofb -e -in f1.txt -out f1-aes-256-ofb.txt -K 00112233445566778889aabbccddeeff -iv 010203040
506070809
[09/20/22]seed@VM:~/.../OFB$ openssl enc -aes-256-ofb -e -in f2.txt -out f2-aes-256-ofb.txt -K 00112233445566778889aabbccddeeff -iv 010203040
506070809
[09/20/22]seed@VM:~/.../OFB$ openssl enc -aes-256-ofb -e -in f3.txt -out f3-aes-256-ofb.txt -K 00112233445566778889aabbccddeeff -iv 010203040
506070809
[09/20/22]seed@VM:~/.../OFB$ openssl enc -aes-256-ofb -d -nopad -in f1-aes-256-ofb.txt -out f1_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 010203040506070809
[09/20/22]seed@VM:~/.../OFB$ openssl enc -aes-256-ofb -d -nopad -in f2-aes-256-ofb.txt -out f2_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 010203040506070809
[09/20/22]seed@VM:~/.../OFB$ openssl enc -aes-256-ofb -d -nopad -in f3-aes-256-ofb.txt -out f3_decrypt.txt -K 00112233445566778889aabbccddeef
f -iv 010203040506070809
[09/20/22]seed@VM:~/.../OFB$ hexdump -C f1_decrypt.txt
00000000  31 32 33 34 35                                    |12345|
00000005
[09/20/22]seed@VM:~/.../OFB$ xxd f1_decrypt.txt
00000000: 3132 3334 35                             12345
[09/20/22]seed@VM:~/.../OFB$ hexdump -C f2_decrypt.txt
00000000  31 32 33 34 35 36 37 38  39 30                    |1234567890|
0000000a
[09/20/22]seed@VM:~/.../OFB$ xxd f2_decrypt.txt
00000000: 3132 3334 3536 3738 3930                 1234567890
[09/20/22]seed@VM:~/.../OFB$ hexdump -C f3_decrypt.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010
[09/20/22]seed@VM:~/.../OFB$ xxd f3_decrypt.txt
00000000: 3132 3334 3536 3738 3930 3132 3334 3536  1234567890123456
[09/20/22]seed@VM:~/.../OFB$ 
```


f1.txt (~/CS5500_Labs/Lab2/Tasks/Task4/OFB) - gedit
Open
Save
f1_decrypt.txt (~/CS5500_Labs/Lab2/Tasks/Task4/OFB
Open
Save

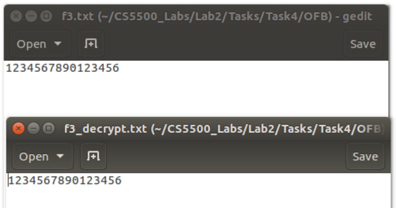
12345

12345


f2.txt (~/CS5500_Labs/Lab2/Tasks/Task4/OFB) - gedit
Open
Save
f2_decrypt.txt (~/CS5500_Labs/Lab2/Tasks/Task4/OFB
Open
Save

1234567890

1234567890

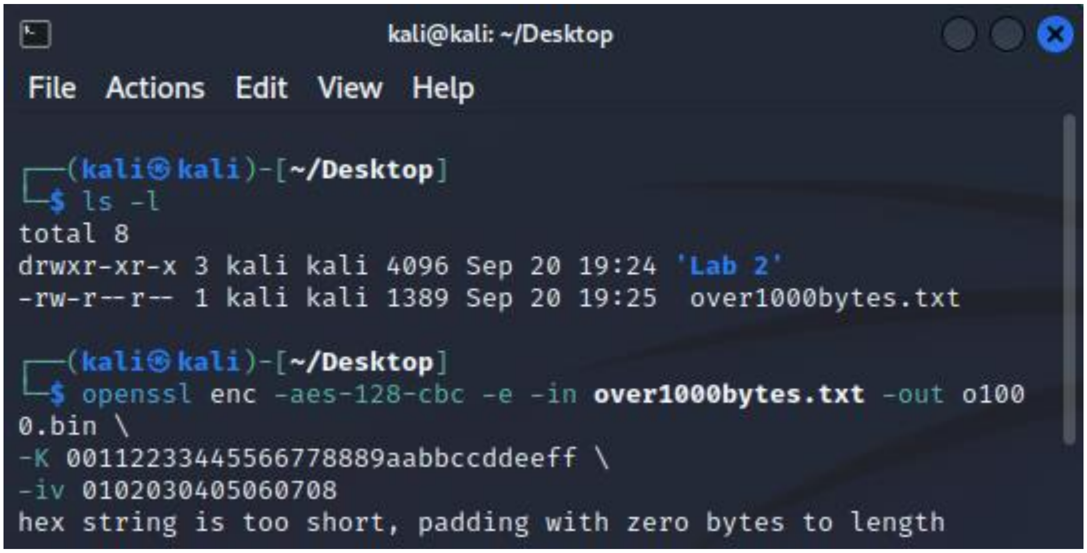

f3.txt (~/CS5500_Labs/Lab2/Tasks/Task4/OFB) - gedit
Open
Save
f3_decrypt.txt (~/CS5500_Labs/Lab2/Tasks/Task4/OFB
Open
Save

1234567890123456

1234567890123456

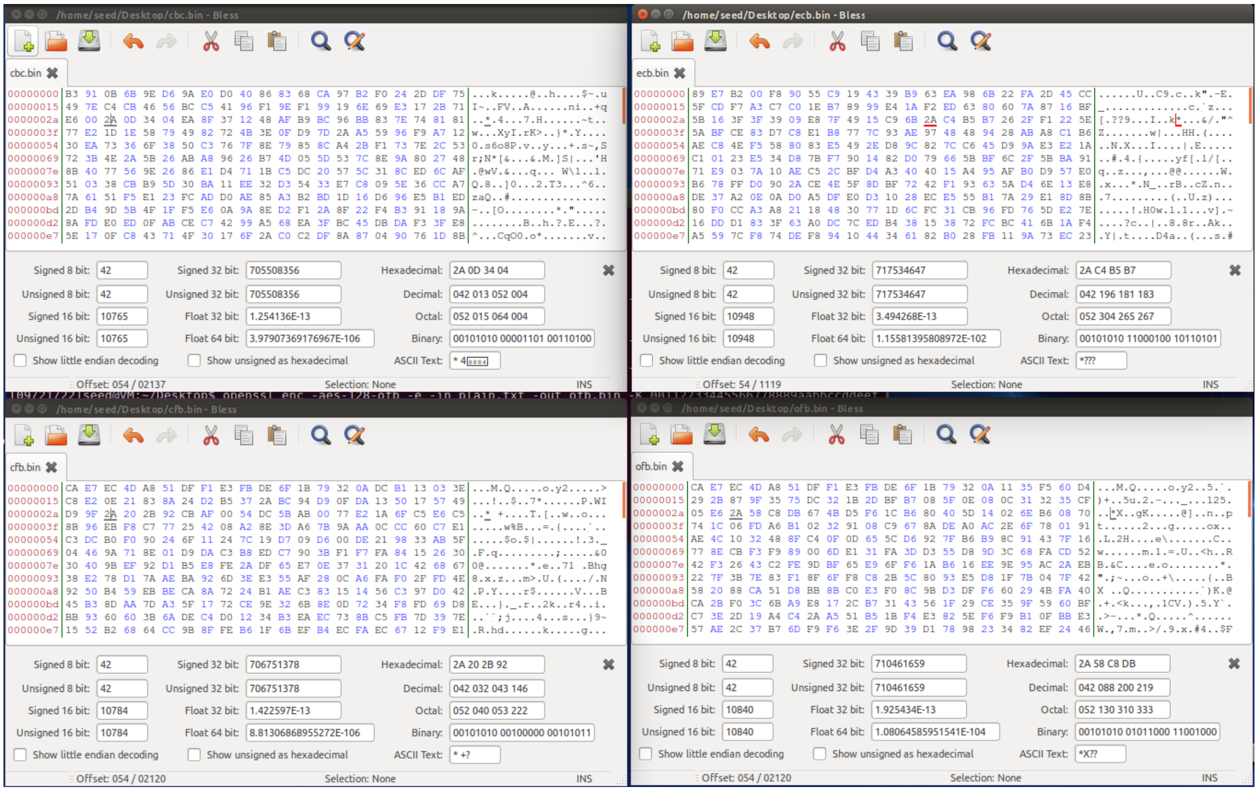
Using OFB (-aes-256-ofb)		
Original Filename	Original Size (Bytes)	Padded Size (Bytes)
f1.txt	5	5
f2.txt	10	10
f3.txt	16	16
OFB is NOT Padded		

Task 5



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ls -l
total 8
drwxr-xr-x 3 kali kali 4096 Sep 20 19:24 'Lab 2'
-rw-r--r-- 1 kali kali 1389 Sep 20 19:25  over1000bytes.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cbc -e -in over1000bytes.txt -out o100
0.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length
```

Corrupt the 55th byte in the encrypted files


/home/seed/Desktop/cbc.bin - Bless
cbc.bin
/home/seed/Desktop/ecb.bin - Bless
ecb.bin
/home/seed/Desktop/cfb.bin - Bless
cfb.bin
/home/seed/Desktop/ofb.bin - Bless
ofb.bin

```
00000000 B3 91 0B 6B 9E D6 9A E0 D0 40 86 83 68 CA 97 B2 F0 24 2D DF 75 ...k.....@..h....$-.u
00000015 49 7E C4 CB 46 56 BC C5 41 96 F1 9E F1 99 19 6E 69 E3 17 2B 71 I~..FV..A......ni..+q
0000002a E6 00 2A 0D 34 04 EA 8F 37 12 48 AF B9 BC 96 BB 83 7E 74 81 81 ..*.4...7.H......~t..
0000003f 77 E2 1D 1E 58 79 49 82 72 4B 3E 0F D9 7D 2A A5 59 96 F9 A7 12 w...XyI.rK>..}*.Y....
00000054 30 EA 73 36 6F 38 50 C3 76 7F 8E 79 85 8C A4 2B F1 73 7E 2C 53 0.s6o8P.v..y...+.s~,S
00000069 72 3B 4E 2A 5B 26 AB A8 96 26 B7 4D 05 5D 53 7C 8E 9A 80 27 48 r;N*[&...&.M.]S|...'H
0000007e 8B 40 77 56 9E 26 86 E1 D4 71 1B C5 DC 20 57 5C 31 8C ED 6C AF .@wV.&...q... W\1..l.
00000093 51 03 38 CB B9 5D 30 BA 11 EE 32 D3 54 33 E7 C8 09 5E 36 CC A7 Q.8..]0...2.T3...^6..
000000a8 7A 61 51 F5 E1 23 FC AD D0 AE 85 A3 B2 BD 1D 16 D6 96 E5 B1 ED zaQ..#...............
000000bd 2D B4 9D 5B 4F 1F F5 E6 0A 9A 8E D2 F1 2A 8F 22 F4 B3 91 18 9A -..[O........*.".....
000000d2 8A FD E0 ED 0F AB CE C7 42 99 A5 68 EA 3F BC 45 DB DA F3 3F E8 ........B..h.?.E...?.
000000e7 5E 17 0F C8 43 71 4F 30 17 6F 2A C0 C2 DF 8A 87 04 90 76 1D 8B ^...CqO0.o*.......v..

Signed 8 bit: 42          Signed 32 bit: 705508356            Hexadecimal: 2A 0D 34 04
Unsigned 8 bit: 42        Unsigned 32 bit: 705508356          Decimal: 042 013 052 004
Signed 16 bit: 10765      Float 32 bit: 1.254136E-13          Octal: 052 015 064 004
Unsigned 16 bit: 10765    Float 64 bit: 3.97907369176967E-106 Binary: 00101010 00001101 00110100
☐ Show little endian decoding   ☐ Show unsigned as hexadecimal   ASCII Text: * 4
Offset: 054 / 02137        Selection: None        INS
```

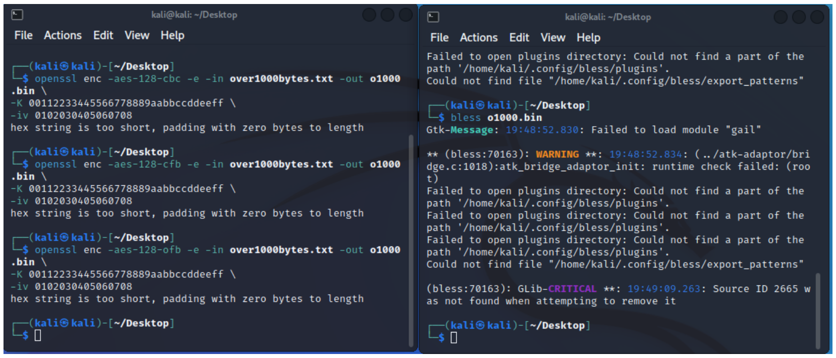
```
00000000 89 E7 B2 00 F8 90 55 C9 19 43 39 B9 63 EA 98 6B 22 FA 2D 45 CC ......U..C9.c..k".-E.
00000015 5F CD F7 A3 C7 C0 1E B7 89 99 E4 1A F2 ED 63 80 60 7A 87 16 BF _............c.`z...
0000002a 5B 16 3F 3F 39 09 E8 7F 49 15 C9 6B 2A C4 B5 B7 26 2F F1 22 5E [.??9...I..k*...&/."^
0000003f 5A BF CE 83 D7 C8 E1 B8 77 7C 93 AE 97 48 48 94 28 AB A8 C1 B6 Z.......w|...HH.(....
00000054 AE C8 4E F5 58 80 83 E5 49 2E D8 9C 82 7C C6 45 D9 9A E3 E2 1A ..N.X...I....|.E.....
00000069 C1 01 23 E5 34 D8 7B F7 90 14 82 D0 79 66 5B BF 6C 2F 5B BA 91 ..#.4.{.....yf[.l/[..
0000007e 71 E9 03 7A 10 AE C5 2C BF D4 A3 40 40 15 A4 95 AF B0 D9 57 E0 q..z...,...@@......W.
00000093 B6 78 FF D0 90 2A CE 4E 5F 8D BF 72 42 F1 93 63 5A D4 6E 13 E8 .x...*.N_..rB..cZ.n..
000000a8 DE 37 A2 0E 0A D0 A5 DF E0 D3 10 28 EC E5 55 B1 7A 29 E1 8D 8B .7.........(..U.z)...
000000bd 80 F0 CC A3 A8 21 18 48 30 77 1D 6C FC 31 CB 96 FD 76 5D E2 7E .....!.H0w.l.1...v].~
000000d2 16 DD D1 83 3F 63 A0 DC 7C ED B4 38 15 38 72 FC BC 41 6B 1A F4 ....?c..|..8.8r..Ak..
000000e7 A5 59 7C F8 74 DE F8 94 10 44 34 61 82 B0 28 FB 11 9A 73 EC 23 .Y|.t....D4a..(...s.#

Signed 8 bit: 42          Signed 32 bit: 717534647            Hexadecimal: 2A C4 B5 B7
Unsigned 8 bit: 42        Unsigned 32 bit: 717534647          Decimal: 042 196 181 183
Signed 16 bit: 10948      Float 32 bit: 3.494268E-13          Octal: 052 304 265 267
Unsigned 16 bit: 10948    Float 64 bit: 1.15581395808972E-102 Binary: 00101010 11000100 10110101
☐ Show little endian decoding   ☐ Show unsigned as hexadecimal   ASCII Text: *???
Offset: 54 / 1119        Selection: None        INS
```

```
00000000 CA E7 EC 4D A8 51 DF F1 E3 FB DE 6F 1B 79 32 0A DC B1 13 03 3E ...M.Q.....o.y2.....>
00000015 C8 E2 0E 21 83 8A 24 D2 B5 37 2A BC 94 D9 0F DA 13 50 17 57 49 ...!..$..7*......P.WI
0000002a D9 9F 2A 20 2B 92 CB AF 00 54 DC 5B AB 00 77 E2 1A 6F C5 E6 C5 ..* +....T.[..w..o...
0000003f 8B 96 EB F8 C7 77 25 42 08 A2 8E 3D A6 7B 9A AA 0C CC 60 C7 E1 .....w%B...=.{....`..
00000054 C3 DC B0 F0 90 24 6F 11 24 7C 19 D7 09 D6 00 DE 21 98 33 AB 5F .....$o.$|......!.3._
00000069 04 46 9A 71 8E 01 D9 DA C3 B8 ED C7 90 3B F1 F7 FA 84 15 26 30 .F.q.........;.....&0
0000007e 30 40 9B EF 92 D1 B5 E8 FE 2A DF 65 E7 0E 37 31 20 1C 42 68 67 0@.......*.e..71 .Bhg
00000093 38 E2 78 D1 7A AE BA 92 6D 3E E3 55 AF 28 0C A6 FA F0 2F FD 4E 8.x.z...m>.U.(..../.N
000000a8 92 50 B4 59 EB BE CA 8A 72 24 B1 AE C3 83 15 14 56 C3 97 D0 42 .P.Y....r$......V...B
000000bd 45 B3 8D AA 7D A3 5F 17 72 CE 9E 32 6B 8E 0D 72 34 F8 FD 69 D8 E...}._.r..2k..r4..i.
000000d2 BB 93 60 60 3B 6A DE C4 D0 12 34 B3 EA EC 73 8B C5 FB 7D 39 7E ..``;j....4...s...}9~
000000e7 15 52 B2 68 64 CC 9B 8F FE B6 1F 6B EF B4 EC FA EC 67 12 F9 E1 .R.hd......k.....g...

Signed 8 bit: 42          Signed 32 bit: 706751378            Hexadecimal: 2A 20 2B 92
Unsigned 8 bit: 42        Unsigned 32 bit: 706751378          Decimal: 042 032 043 146
Signed 16 bit: 10784      Float 32 bit: 1.422597E-13          Octal: 052 040 053 222
Unsigned 16 bit: 10784    Float 64 bit: 8.81306868955272E-106 Binary: 00101010 00100000 00101011
☐ Show little endian decoding   ☐ Show unsigned as hexadecimal   ASCII Text: * +?
Offset: 054 / 02120        Selection: None        INS
```

```
00000000 CA E7 EC 4D A8 51 DF F1 E3 FB DE 6F 1B 79 32 0A 11 35 F5 60 D4 ...M.Q.....o.y2..5.`.
00000015 29 2B 87 9F 35 75 DC 32 1B 2D BF B7 08 5F 0E 08 0C 31 32 35 CF )+..5u.2.-..._...125.
0000002a 05 E6 2A 58 C8 DB 67 4B D5 F6 1C B6 80 40 5D 14 02 6E B6 08 70 ..*X..gK.....@]..n..p
0000003f 74 1C 06 FD A6 B1 02 32 91 08 C9 67 8A DE A0 AC 2E 6F 78 01 91 t......2...g.....ox..
00000054 AE 4C 10 32 48 8F C4 0F 0D 65 5C D6 92 7F B6 B9 8C 91 43 7F 16 .L.2H....e\.......C..
00000069 77 8E CB F3 F9 89 00 6D E1 31 FA 3D D3 55 D8 9D 3C 68 FA CD 52 w......m.1.=.U..<h..R
0000007e 42 F3 26 43 C2 FE 9D BF 65 E9 6F F6 1A B6 16 EE 9E 95 AC 2A EB B.&C....e.o.........*.
00000093 22 7F 3B 7E 83 F1 8F 6F F8 C8 2B 5C 80 93 E5 D8 1F 7B 04 7F 42 ".;~...o..+\.....{..B
000000a8 58 20 88 CA 51 D8 BB 8B C0 E3 F0 8C 9B D3 DF F6 60 29 4B FA 40 X ..Q...........`)K.@
000000bd CA 2B F0 3C 6B A9 E8 17 2C B7 31 43 56 1F 29 CE 35 9F 59 60 BF .+.<k...,.1CV.).5.Y`.
000000d2 C7 3E 2D 19 A4 C4 2A A5 51 B5 1B F4 E3 82 5E F6 F9 B1 0F BB E3 .>-...*.Q.....^......
000000e7 57 AE 2C 37 B7 6D F9 F6 3E 2F 9D 39 D1 78 98 23 34 82 EF 24 46 W.,7.m..>/.9.x.#4..$F

Signed 8 bit: 42          Signed 32 bit: 710461659            Hexadecimal: 2A 58 C8 DB
Unsigned 8 bit: 42        Unsigned 32 bit: 710461659          Decimal: 042 088 200 219
Signed 16 bit: 10840      Float 32 bit: 1.925434E-13          Octal: 052 130 310 333
Unsigned 16 bit: 10840    Float 64 bit: 1.08064585951541E-104 Binary: 00101010 01011000 11001000
☐ Show little endian decoding   ☐ Show unsigned as hexadecimal   ASCII Text: *X??
Offset: 054 / 02120        Selection: None        INS
```

XORing changes a single bit if a 1 is used with several 0s.

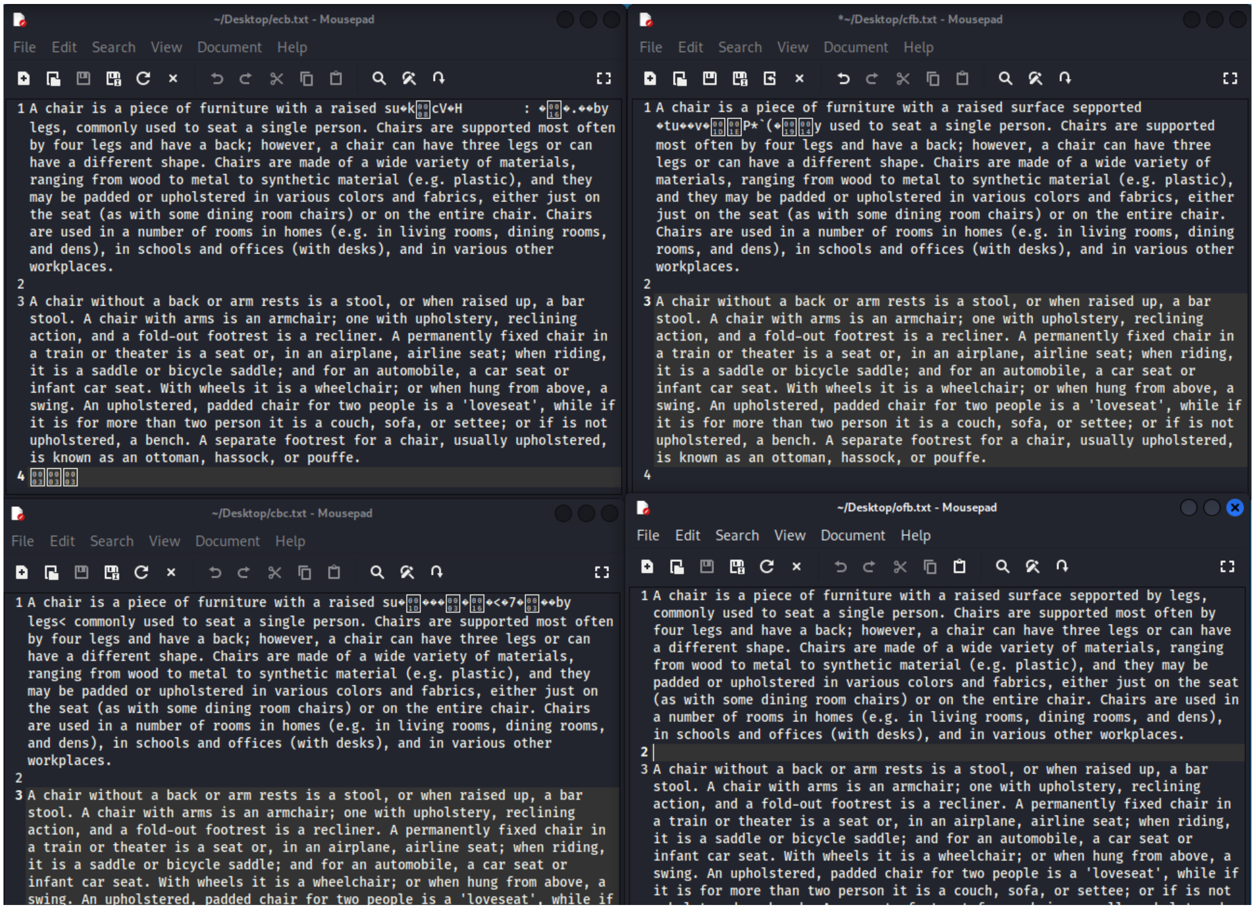
How much information can you recover by decrypting the corrupted file, if the encryption mode is ECB, CBC, CFB, or OFB, respectively? Please answer this question before you conduct this task, and then find out whether your answer is correct or wrong after you finish this task. Please provide justification.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cbc -e -in over1000bytes.txt -out o1000
.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-cfb -e -in over1000bytes.txt -out o1000
.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ openssl enc -aes-128-ofb -e -in over1000bytes.txt -out o1000
.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708
hex string is too short, padding with zero bytes to length

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

```
Failed to open plugins directory: Could not find a part of the
path '/home/kali/.config/bless/plugins'.
Could not find file "/home/kali/.config/bless/export_patterns"

┌──(kali㉿kali)-[~/Desktop]
└─$ bless o1000.bin
Gtk-Message: 19:48:52.830: Failed to load module "gail"

** (bless:70163): WARNING **: 19:48:52.834: (../atk-adaptor/bri
dge.c:1018):atk_bridge_adaptor_init: runtime check failed: (roo
t)
Failed to open plugins directory: Could not find a part of the
path '/home/kali/.config/bless/plugins'.
Failed to open plugins directory: Could not find a part of the
path '/home/kali/.config/bless/plugins'.
Failed to open plugins directory: Could not find a part of the
path '/home/kali/.config/bless/plugins'.
Could not find file "/home/kali/.config/bless/export_patterns"

(bless:70163): GLib-CRITICAL **: 19:49:09.263: Source ID 2665 w
as not found when attempting to remove it

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

Since CBC and CFB encryption are not based on the text within the document, we would expect most of the text to be retrievable (other than the corrupted blocks). I expect ECB and OFB to be similar as well.



~/Desktop/ecb.txt - Mousepad
File Edit Search View Document Help

```
1 A chair is a piece of furniture with a raised su�k□cV�H       : �□�.��by
  legs, commonly used to seat a single person. Chairs are supported most often
  by four legs and have a back; however, a chair can have three legs or can
  have a different shape. Chairs are made of a wide variety of materials,
  ranging from wood to metal to synthetic material (e.g. plastic), and they
  may be padded or upholstered in various colors and fabrics, either just on
  the seat (as with some dining room chairs) or on the entire chair. Chairs
  are used in a number of rooms in homes (e.g. in living rooms, dining rooms,
  and dens), in schools and offices (with desks), and in various other
  workplaces.
2
3 A chair without a back or arm rests is a stool, or when raised up, a bar
  stool. A chair with arms is an armchair; one with upholstery, reclining
  action, and a fold-out footrest is a recliner. A permanently fixed chair in
  a train or theater is a seat or, in an airplane, airline seat; when riding,
  it is a saddle or bicycle saddle; and for an automobile, a car seat or
  infant car seat. With wheels it is a wheelchair; or when hung from above, a
  swing. An upholstered, padded chair for two people is a 'loveseat', while if
  it is for more than two person it is a couch, sofa, or settee; or if is not
  upholstered, a bench. A separate footrest for a chair, usually upholstered,
  is known as an ottoman, hassock, or pouffe.
4 □□□
```

*~/Desktop/cfb.txt - Mousepad
File Edit Search View Document Help

```
1 A chair is a piece of furniture with a raised surface sepported
  �tu��v�□□P*`(�□□y used to seat a single person. Chairs are supported
  most often by four legs and have a back; however, a chair can have three
  legs or can have a different shape. Chairs are made of a wide variety of
  materials, ranging from wood to metal to synthetic material (e.g. plastic),
  and they may be padded or upholstered in various colors and fabrics, either
  just on the seat (as with some dining room chairs) or on the entire chair.
  Chairs are used in a number of rooms in homes (e.g. in living rooms, dining
  rooms, and dens), in schools and offices (with desks), and in various other
  workplaces.
2
3 A chair without a back or arm rests is a stool, or when raised up, a bar
  stool. A chair with arms is an armchair; one with upholstery, reclining
  action, and a fold-out footrest is a recliner. A permanently fixed chair in
  a train or theater is a seat or, in an airplane, airline seat; when riding,
  it is a saddle or bicycle saddle; and for an automobile, a car seat or
  infant car seat. With wheels it is a wheelchair; or when hung from above, a
  swing. An upholstered, padded chair for two people is a 'loveseat', while if
  it is for more than two person it is a couch, sofa, or settee; or if is not
  upholstered, a bench. A separate footrest for a chair, usually upholstered,
  is known as an ottoman, hassock, or pouffe.
4
```

~/Desktop/cbc.txt - Mousepad
File Edit Search View Document Help

```
1 A chair is a piece of furniture with a raised su�□���□�□�<�7�□��by
  legs< commonly used to seat a single person. Chairs are supported most often
  by four legs and have a back; however, a chair can have three legs or can
  have a different shape. Chairs are made of a wide variety of materials,
  ranging from wood to metal to synthetic material (e.g. plastic), and they
  may be padded or upholstered in various colors and fabrics, either just on
  the seat (as with some dining room chairs) or on the entire chair. Chairs
  are used in a number of rooms in homes (e.g. in living rooms, dining rooms,
  and dens), in schools and offices (with desks), and in various other
  workplaces.
2
3 A chair without a back or arm rests is a stool, or when raised up, a bar
  stool. A chair with arms is an armchair; one with upholstery, reclining
  action, and a fold-out footrest is a recliner. A permanently fixed chair in
  a train or theater is a seat or, in an airplane, airline seat; when riding,
  it is a saddle or bicycle saddle; and for an automobile, a car seat or
  infant car seat. With wheels it is a wheelchair; or when hung from above, a
  swing. An upholstered, padded chair for two people is a 'loveseat', while if
```

~/Desktop/ofb.txt - Mousepad
File Edit Search View Document Help

```
1 A chair is a piece of furniture with a raised surface sepported by legs,
  commonly used to seat a single person. Chairs are supported most often by
  four legs and have a back; however, a chair can have three legs or can have
  a different shape. Chairs are made of a wide variety of materials, ranging
  from wood to metal to synthetic material (e.g. plastic), and they may be
  padded or upholstered in various colors and fabrics, either just on the seat
  (as with some dining room chairs) or on the entire chair. Chairs are used in
  a number of rooms in homes (e.g. in living rooms, dining rooms, and dens),
  in schools and offices (with desks), and in various other workplaces.
2
3 A chair without a back or arm rests is a stool, or when raised up, a bar
  stool. A chair with arms is an armchair; one with upholstery, reclining
  action, and a fold-out footrest is a recliner. A permanently fixed chair in
  a train or theater is a seat or, in an airplane, airline seat; when riding,
  it is a saddle or bicycle saddle; and for an automobile, a car seat or
  infant car seat. With wheels it is a wheelchair; or when hung from above, a
  swing. An upholstered, padded chair for two people is a 'loveseat', while if
  it is for more than two person it is a couch, sofa, or settee; or if is not
```

Surprisingly, it appears that very little corruption existed in any document, and (perhaps more surprisingly) the OFB document appears not to have changed at all. It seems that all four encryption methods work well even with some amount of file corruption.