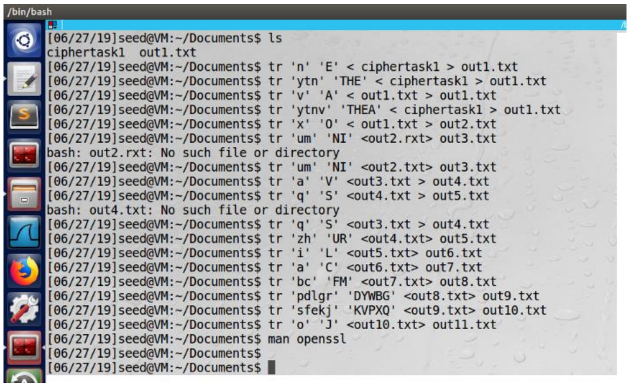
Secret-key encryption
Substitution cipher and frequency analysis
Encryption modes and paddings
Programming using the crypto library

2.1 Task 1:

A ***monoalphabetic substitution cipher***, also known as a simple substitution cipher, relies on a fixed replacement structure. That is, the substitution is fixed for each letter of the alphabet. Thus, if "a" is encrypted to "R", then every time we see the letter "a" in the plaintext, we replace it with the letter "R" in the ciphertext.
https://crypto.interactive-maths.com/monoalphabetic-substitution-ciphers.html



```
[06/27/19]seed@VM:~/Documents$ ls
ciphertask1  out1.txt
[06/27/19]seed@VM:~/Documents$ tr 'n' 'E' < ciphertask1 > out1.txt
[06/27/19]seed@VM:~/Documents$ tr 'ytn' 'THE' < ciphertask1 > out1.txt
[06/27/19]seed@VM:~/Documents$ tr 'v' 'A' < out1.txt > out1.txt
[06/27/19]seed@VM:~/Documents$ tr 'ytnv' 'THEA' < ciphertask1 > out1.txt
[06/27/19]seed@VM:~/Documents$ tr 'x' 'O' < out1.txt > out2.txt
[06/27/19]seed@VM:~/Documents$ tr 'um' 'NI' <out2.rxt> out3.txt
bash: out2.rxt: No such file or directory
[06/27/19]seed@VM:~/Documents$ tr 'um' 'NI' <out2.txt> out3.txt
[06/27/19]seed@VM:~/Documents$ tr 'a' 'V' <out3.txt > out4.txt
[06/27/19]seed@VM:~/Documents$ tr 'q' 'S' <out4.txt > out5.txt
bash: out4.txt: No such file or directory
[06/27/19]seed@VM:~/Documents$ tr 'q' 'S' <out3.txt > out4.txt
[06/27/19]seed@VM:~/Documents$ tr 'zh' 'UR' <out4.txt> out5.txt
[06/27/19]seed@VM:~/Documents$ tr 'i' 'L' <out5.txt> out6.txt
[06/27/19]seed@VM:~/Documents$ tr 'a' 'C' <out6.txt> out7.txt
[06/27/19]seed@VM:~/Documents$ tr 'bc' 'FM' <out7.txt> out8.txt
[06/27/19]seed@VM:~/Documents$ tr 'pdlgr' 'DYWBG' <out8.txt> out9.txt
[06/27/19]seed@VM:~/Documents$ tr 'sfekj' 'KVPXQ' <out9.txt> out10.txt
[06/27/19]seed@VM:~/Documents$ tr 'o' 'J' <out10.txt> out11.txt
[06/27/19]seed@VM:~/Documents$ man openssl
[06/27/19]seed@VM:~/Documents$
[06/27/19]seed@VM:~/Documents$ 
```

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

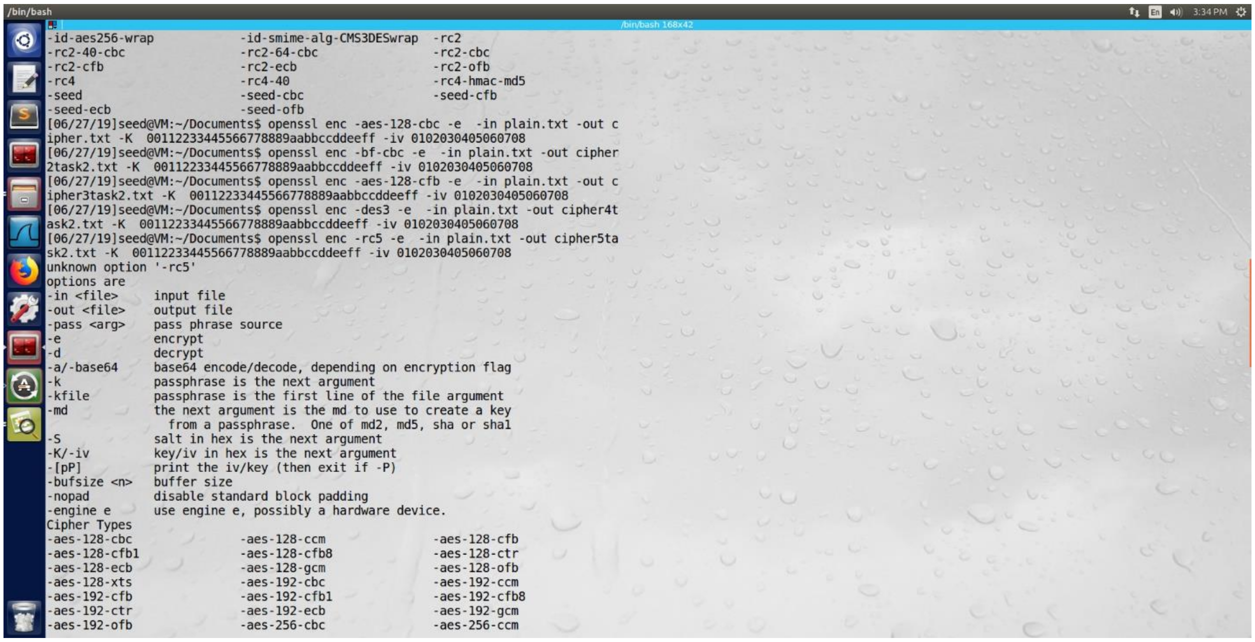
IN IT WAS A TOSSUP BETWEEN BOYHOOD AND THE EVENTUAL WINNER BIRDMAN IN WITH LOTS OF EXPERTS BETTING ON THE REVENANT OR THE BIG SHORT THE PRIwE WENT TO SPOTLIGHT LAST YEAR NEARLY ALL THE FORECASTERS DECLARED LA LA LAND THE PRESUMPTIVE WINNER AND FOR TWO AND A HALF MINUTES THEY WERE CORRECT BEFORE AN ENVELOPE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

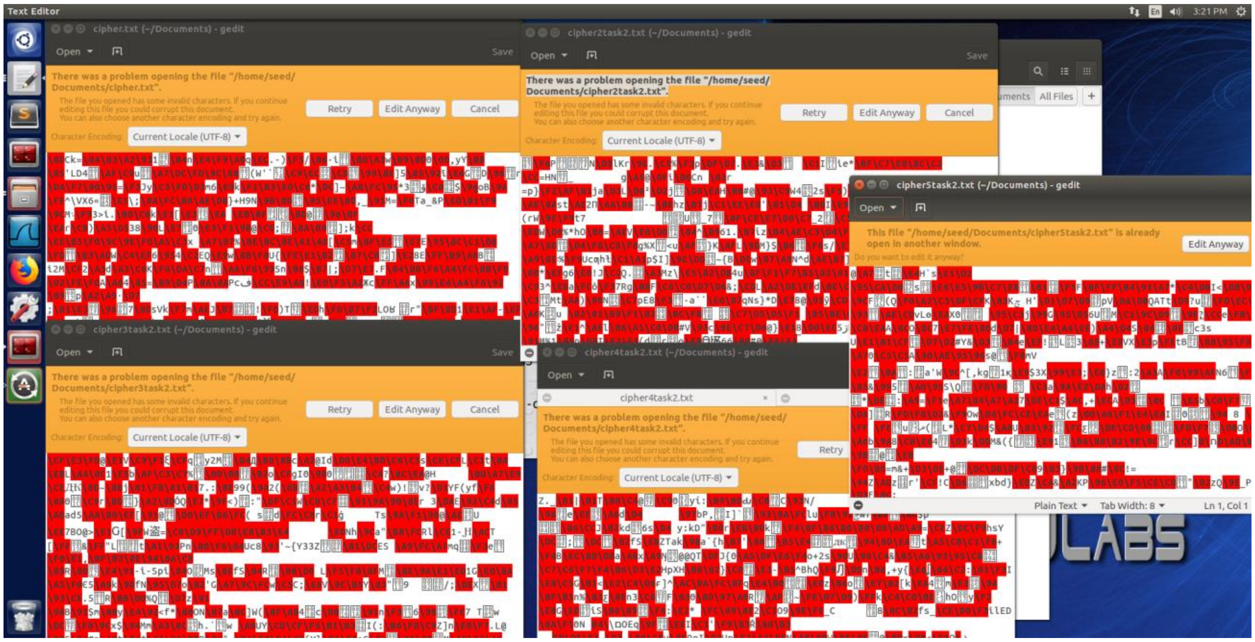
BUT ALL OF THOSE FILMS HAVE HISTORICAL OSCARVOTING PATTERNS AGAINST THEM THE SHAPE OF WATER HAS NOMINATIONS MORE THAN ANY OTHER FILM AND WAS ALSO NAMED THE YEARS BEST BY THE PRODUCERS AND DIRECTORS GUILDS YET IT WAS NOT NOMINATED FOR A SCREEN ACTORS GUILD AWARD FOR BEST ENSEMBLE AND NO FILM HAS WON BEST 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

2.2 Task 2:

The encryption methods used were: AES 128bit in CBC, Blowfish in CBC, AES 128bit in CFB, RC4 128 bit, and des-ede3-cbc.



```
-id-aes256-wrap            -id-smime-alg-CMS3DESwrap  -rc2
-rc2-40-cbc                -rc2-64-cbc                -rc2-cbc
-rc2-cfb                   -rc2-ecb                   -rc2-ofb
-rc4                       -rc4-40                    -rc4-hmac-md5
-seed                      -seed-cbc                  -seed-cfb
-seed-ecb                  -seed-ofb
[06/27/19]seed@VM:~/Documents$ openssl enc -aes-128-cbc -e  -in plain.txt -out c
ipher.txt -K  00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/Documents$ openssl enc -bf-cbc -e  -in plain.txt -out cipher
2task2.txt -K  00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/Documents$ openssl enc -aes-128-cfb -e  -in plain.txt -out c
ipher3task2.txt -K  00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/Documents$ openssl enc -des3 -e  -in plain.txt -out cipher4t
ask2.txt -K  00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/Documents$ openssl enc -rc5 -e  -in plain.txt -out cipher5ta
sk2.txt -K  00112233445566778889aabbccddeeff -iv 0102030405060708
unknown option '-rc5'
options are
-in <file>     input file
-out <file>    output file
-pass <arg>    pass phrase source
-e             encrypt
-d             decrypt
-a/-base64     base64 encode/decode, depending on encryption flag
-k             passphrase is the next argument
-kfile         passphrase is the first line of the file argument
-md            the next argument is the md to use to create a key
                 from a passphrase.  One of md2, md5, sha or sha1
-S             salt in hex is the next argument
-K/-iv         key/iv in hex is the next argument
-[pP]          print the iv/key (then exit if -P)
-bufsize <n>   buffer size
-nopad         disable standard block padding
-engine e      use engine e, possibly a hardware device.
Cipher Types
-aes-128-cbc               -aes-128-ccm               -aes-128-cfb
-aes-128-cfb1              -aes-128-cfb8              -aes-128-ctr
-aes-128-ecb               -aes-128-gcm               -aes-128-ofb
-aes-128-xts               -aes-192-cbc               -aes-192-ccm
-aes-192-cfb               -aes-192-cfb1              -aes-192-cfb8
-aes-192-ctr               -aes-192-ecb               -aes-192-gcm
-aes-192-ofb               -aes-256-cbc               -aes-256-ccm
```



Text Editor
3:21 PM
cipher.txt (~/Documents) - gedit
cipher2task2.txt (~/Documents) - gedit
cipher3task2.txt (~/Documents) - gedit
cipher4task2.txt (~/Documents) - gedit
cipher5task2.txt (~/Documents) - gedit
Open
Save
Retry
Edit Anyway
Cancel
Plain Text
Tab Width: 8
Ln 1, Col 1

There was a problem opening the file "/home/seed/Documents/cipher.txt".
The file you opened has some invalid characters. If you continue editing this file you could corrupt this document.
You can also choose another character encoding and try again.
Character Encoding: Current Locale (UTF-8)

There was a problem opening the file "/home/seed/Documents/cipher2task2.txt".

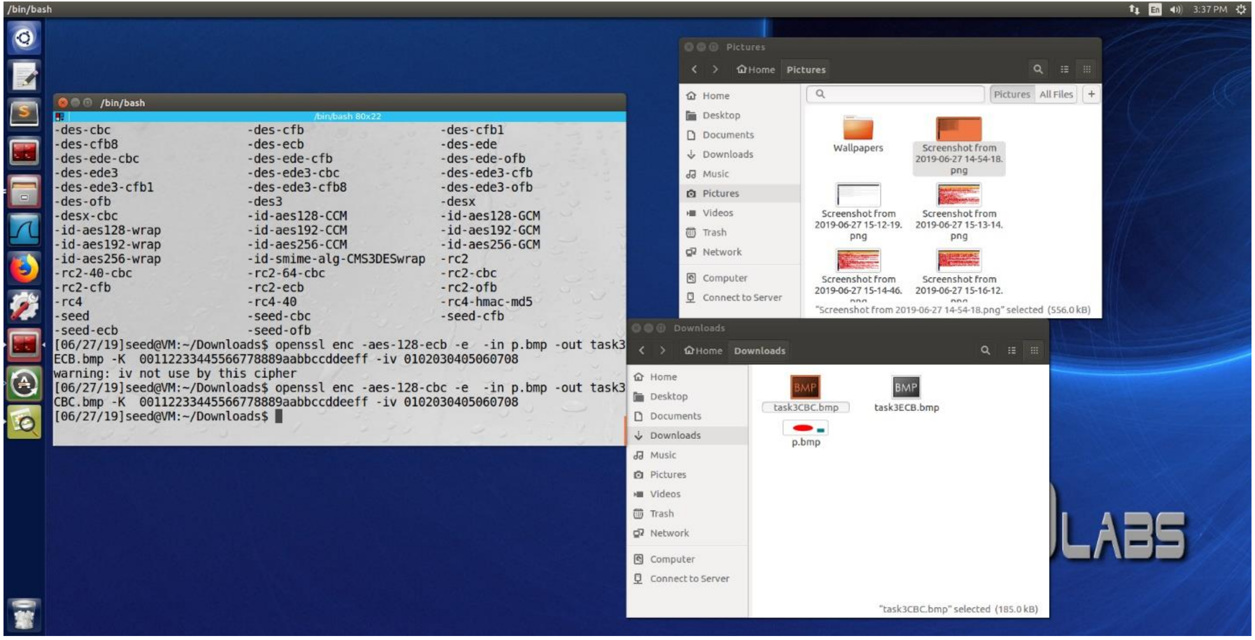
There was a problem opening the file "/home/seed/Documents/cipher3task2.txt".

There was a problem opening the file "/home/seed/Documents/cipher4task2.txt".

This file "/home/seed/Documents/cipher5task2.txt" is already open in another window.
Do you want to edit it anyway?

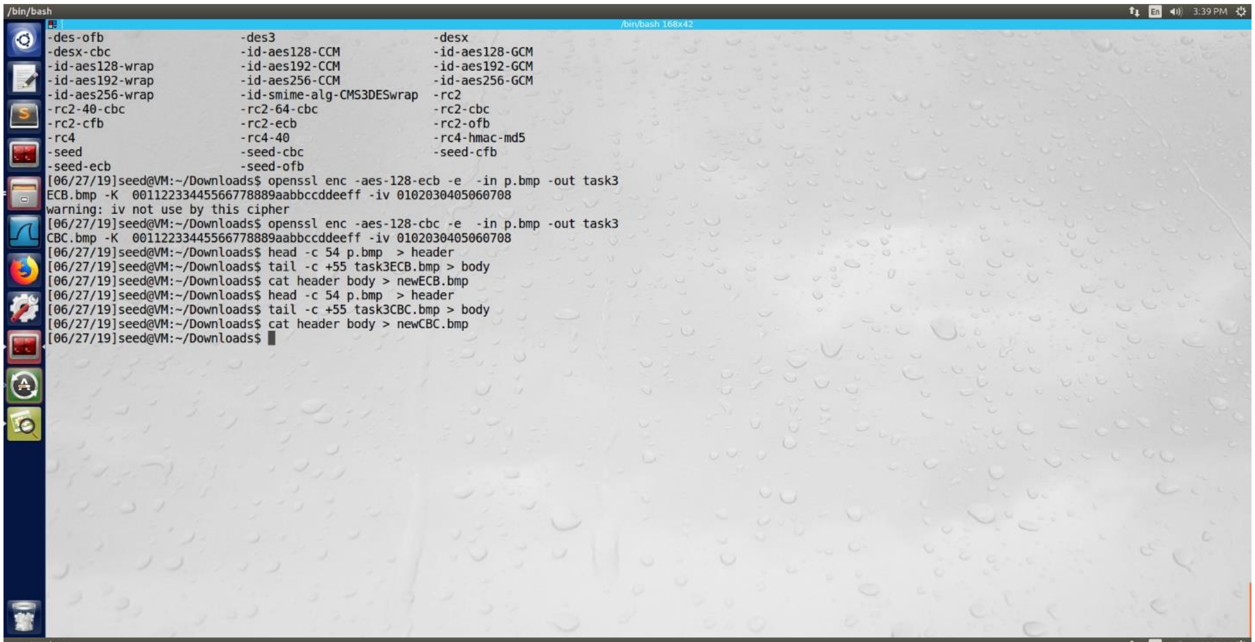
2.3 Task 3:

The ECB picture (newECB.bmp) did share some resemble to that of the original picture. It appears to have flipped the red and blue colors and fuzz up the picture some, but the outline of the picture was distinctly the same. The CBC picture (newCBC.bmp) wasn't recognizable at all. It looked like it snow crashed.

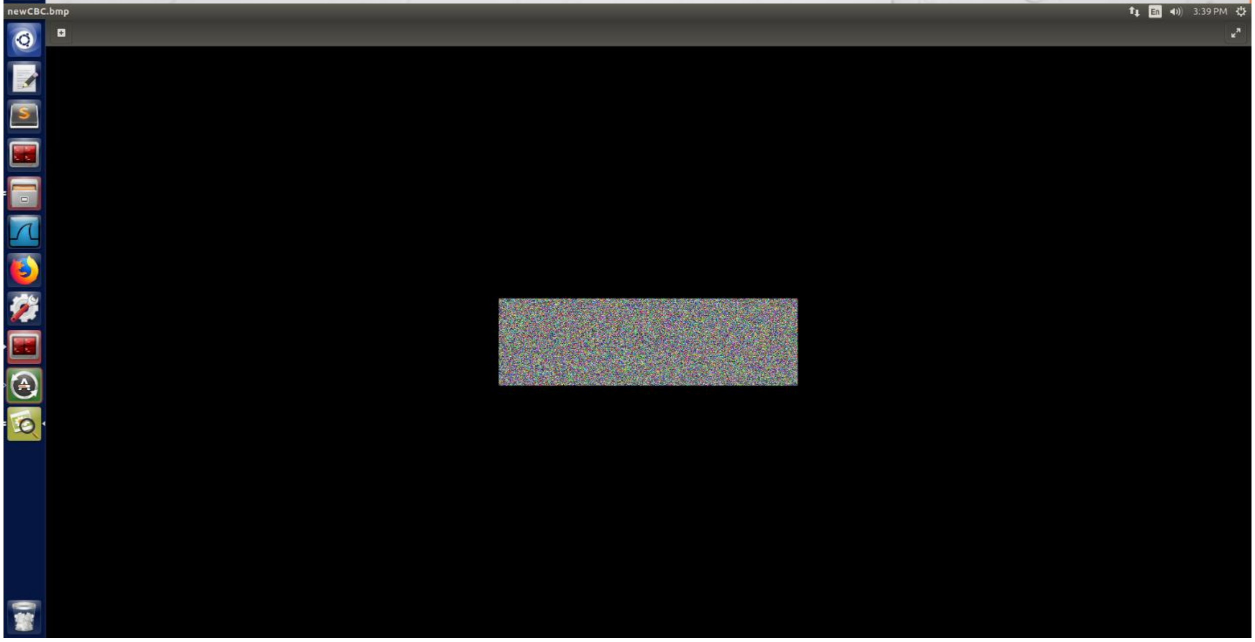


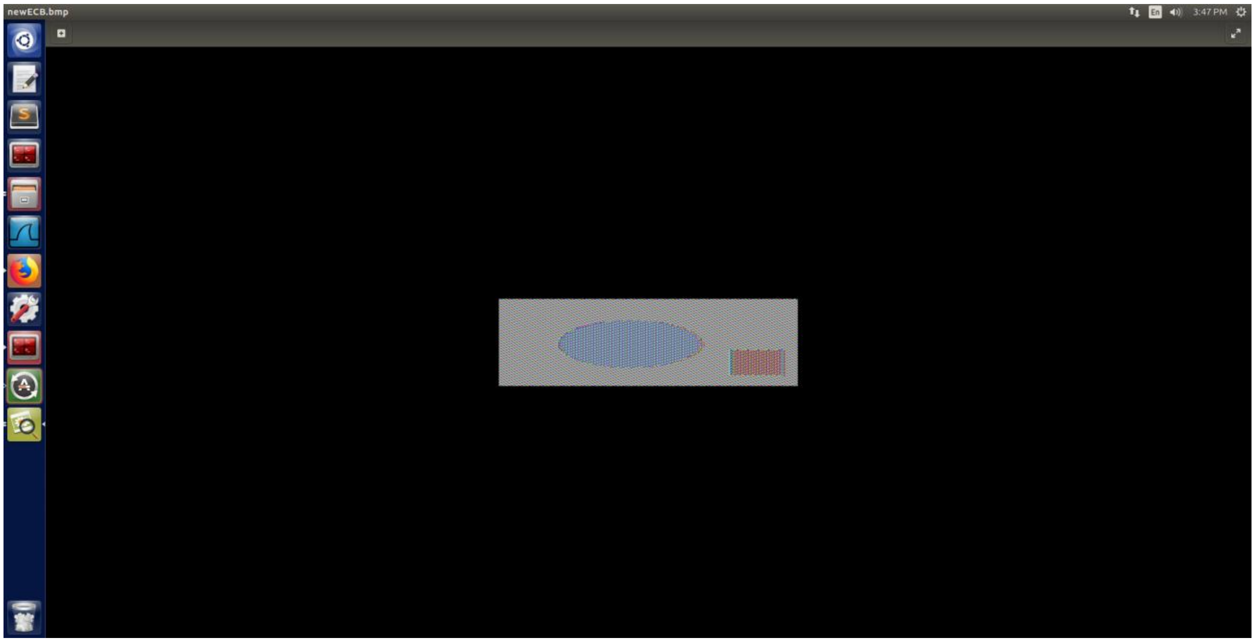
/bin/bash
3:37 PM
/bin/bash 80x22
Pictures
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
Connect to Server
Wallpapers
Screenshot from 2019-06-27 14-54-18.png
Screenshot from 2019-06-27 15-12-19.png
Screenshot from 2019-06-27 15-13-14.png
Screenshot from 2019-06-27 15-14-46.png
Screenshot from 2019-06-27 15-16-12.png
"Screenshot from 2019-06-27 14-54-18.png" selected (556.0 kB)
All Files
task3CBC.bmp
task3ECB.bmp
p.bmp
"task3CBC.bmp" selected (185.0 kB)

```
-des-cbc                    -des-cfb                    -des-cfb1
-des-cfb8                   -des-ecb                    -des-ede
-des-ede-cbc                -des-ede-cfb                -des-ede-ofb
-des-ede3                   -des-ede3-cbc               -des-ede3-cfb
-des-ede3-cfb1              -des-ede3-cfb8              -des-ede3-ofb
-des-ofb                    -des3                       -desx
-desx-cbc                   -id-aes128-CCM              -id-aes128-GCM
-id-aes128-wrap             -id-aes192-CCM              -id-aes192-GCM
-id-aes192-wrap             -id-aes256-CCM              -id-aes256-GCM
-id-aes256-wrap             -id-smime-alg-CMS3DESwrap   -rc2
-rc2-40-cbc                 -rc2-64-cbc                 -rc2-cbc
-rc2-cfb                    -rc2-ecb                    -rc2-ofb
-rc4                        -rc4-40                     -rc4-hmac-md5
-seed                       -seed-cbc                   -seed-cfb
-seed-ecb                   -seed-ofb
[06/27/19]seed@VM:~/Downloads$ openssl enc -aes-128-ecb -e  -in p.bmp -out task3
ECB.bmp -K  00112233445566778889aabbccddeeff -iv 0102030405060708
warning: iv not use by this cipher
[06/27/19]seed@VM:~/Downloads$ openssl enc -aes-128-cbc -e  -in p.bmp -out task3
CBC.bmp -K  00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/Downloads$ 
```

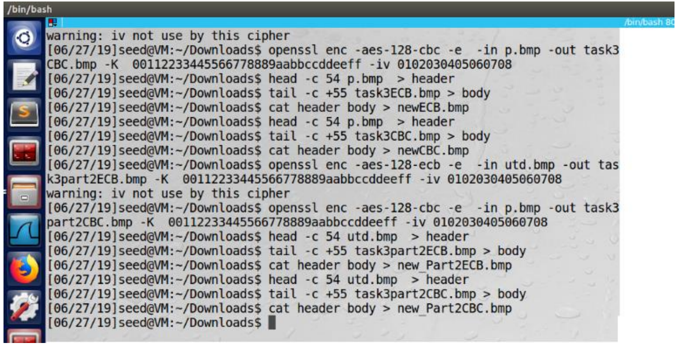


```
-des-ofb                   -des3                      -desx
-desx-cbc                  -id-aes128-CCM             -id-aes128-GCM
-id-aes128-wrap            -id-aes192-CCM             -id-aes192-GCM
-id-aes192-wrap            -id-aes256-CCM             -id-aes256-GCM
-id-aes256-wrap            -id-smime-alg-CMS3DESwrap  -rc2
-rc2-40-cbc                -rc2-64-cbc                -rc2-cbc
-rc2-cfb                   -rc2-ecb                   -rc2-ofb
-rc4                       -rc4-40                    -rc4-hmac-md5
-seed                      -seed-cbc                  -seed-cfb
-seed-ecb                  -seed-ofb
[06/27/19]seed@VM:~/Downloads$ openssl enc -aes-128-ecb -e  -in p.bmp -out task3
ECB.bmp -K  00112233445566778889aabbccddeeff -iv 0102030405060708
warning: iv not use by this cipher
[06/27/19]seed@VM:~/Downloads$ openssl enc -aes-128-cbc -e  -in p.bmp -out task3
CBC.bmp -K  00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/Downloads$ head -c 54 p.bmp  > header
[06/27/19]seed@VM:~/Downloads$ tail -c +55 task3ECB.bmp > body
[06/27/19]seed@VM:~/Downloads$ cat header body > newECB.bmp
[06/27/19]seed@VM:~/Downloads$ head -c 54 p.bmp  > header
[06/27/19]seed@VM:~/Downloads$ tail -c +55 task3CBC.bmp > body
[06/27/19]seed@VM:~/Downloads$ cat header body > newCBC.bmp
[06/27/19]seed@VM:~/Downloads$ █
```





I didn't notice anything from the picture I selected. No similarities revealed between the encrypted and non-encrypted picture.



```
warning: iv not use by this cipher
[06/27/19]seed@VM:~/Downloads$ openssl enc -aes-128-cbc -e  -in p.bmp -out task3
CBC.bmp -K  00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/Downloads$ head -c 54 p.bmp  > header
[06/27/19]seed@VM:~/Downloads$ tail -c +55 task3ECB.bmp > body
[06/27/19]seed@VM:~/Downloads$ cat header body > newECB.bmp
[06/27/19]seed@VM:~/Downloads$ head -c 54 p.bmp  > header
[06/27/19]seed@VM:~/Downloads$ tail -c +55 task3CBC.bmp > body
[06/27/19]seed@VM:~/Downloads$ cat header body > newCBC.bmp
[06/27/19]seed@VM:~/Downloads$ openssl enc -aes-128-ecb -e  -in utd.bmp -out tas
k3part2ECB.bmp -K  00112233445566778889aabbccddeeff -iv 0102030405060708
warning: iv not use by this cipher
[06/27/19]seed@VM:~/Downloads$ openssl enc -aes-128-cbc -e  -in p.bmp -out task3
part2CBC.bmp -K  00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/Downloads$ head -c 54 utd.bmp  > header
[06/27/19]seed@VM:~/Downloads$ tail -c +55 task3part2ECB.bmp > body
[06/27/19]seed@VM:~/Downloads$ cat header body > new_Part2ECB.bmp
[06/27/19]seed@VM:~/Downloads$ head -c 54 utd.bmp  > header
[06/27/19]seed@VM:~/Downloads$ tail -c +55 task3part2CBC.bmp > body
[06/27/19]seed@VM:~/Downloads$ cat header body > new_Part2CBC.bmp
[06/27/19]seed@VM:~/Downloads$ █
```

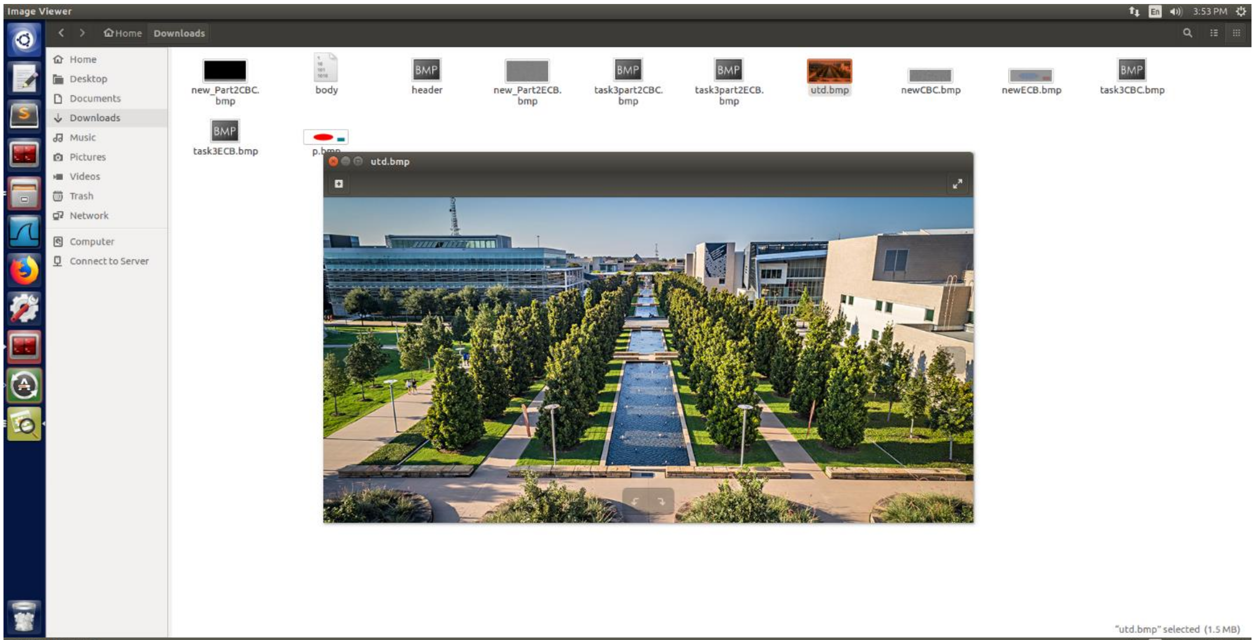
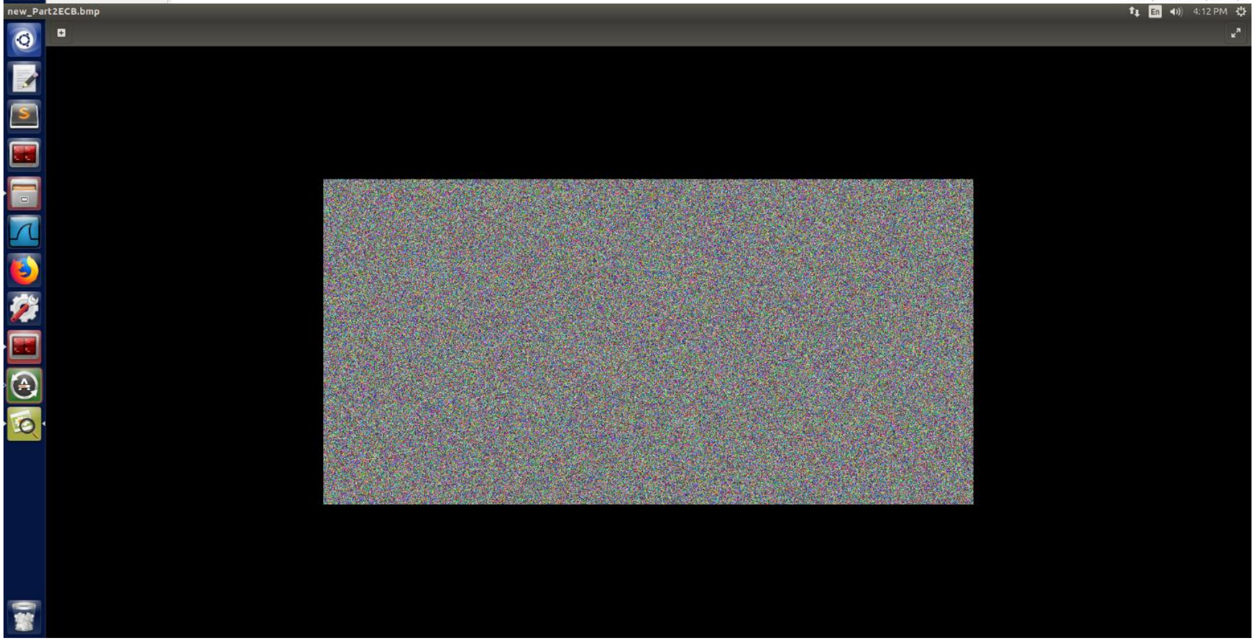
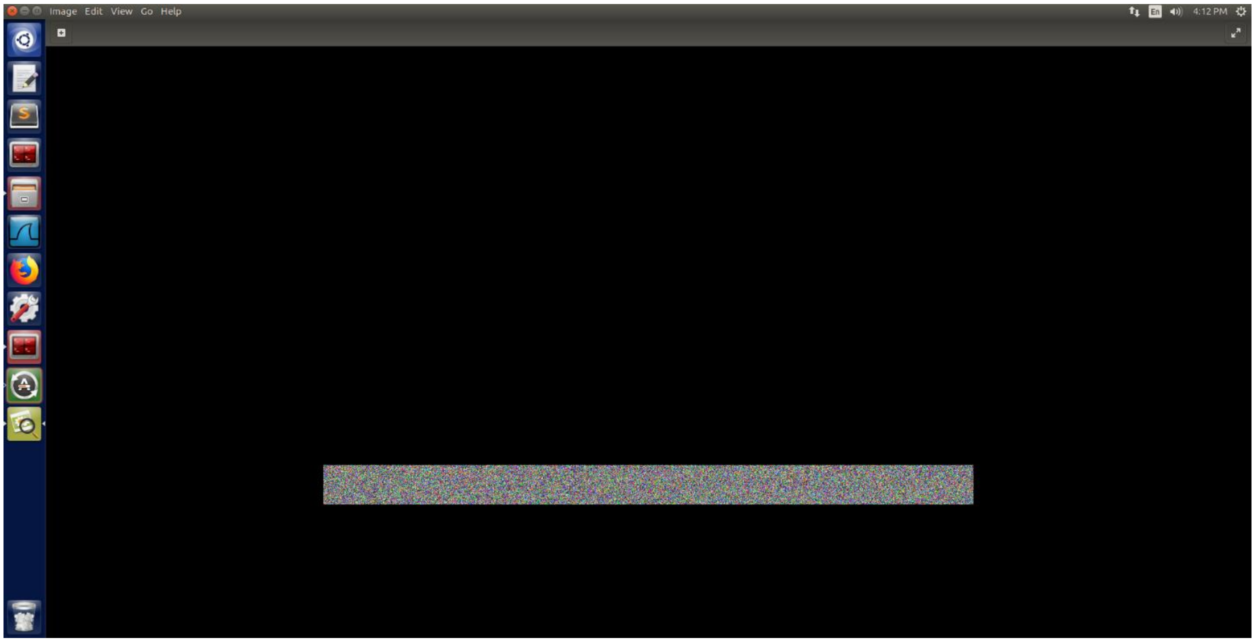


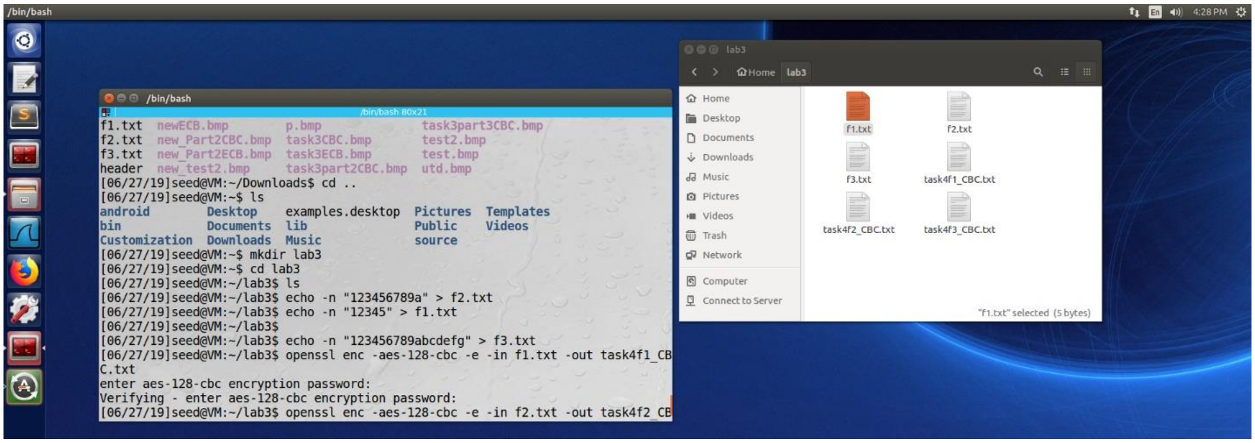
Image Viewer
3:53 PM
Home
Downloads
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
Connect to Server
new_Part2CBC.bmp
body
header
new_Part2ECB.bmp
task3part2CBC.bmp
task3part2ECB.bmp
utd.bmp
newCBC.bmp
newECB.bmp
task3CBC.bmp
task3ECB.bmp
p.bmp
utd.bmp
"utd.bmp" selected (1.5 MB)


new_Part2ECB.bmp
4:12 PM



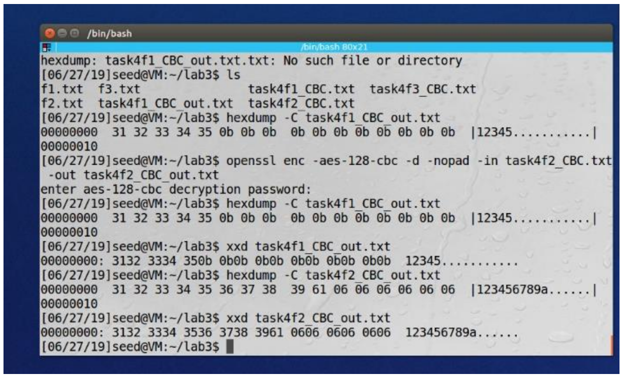
Above is the original picture and encrypted pictures.

2.4 Task4:

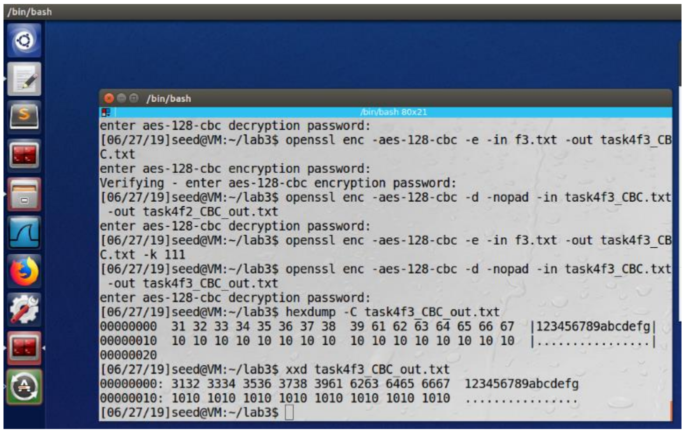


```
f1.txt  newECB.bmp        p.bmp              task3part3CBC.bmp
f2.txt  new_Part2CBC.bmp  task3CBC.bmp       test2.bmp
f3.txt  new_Part2ECB.bmp  task3ECB.bmp       test.bmp
header  new_test2.bmp     task3part2CBC.bmp  utd.bmp
[06/27/19]seed@VM:~/Downloads$ cd ..
[06/27/19]seed@VM:~$ ls
android        Desktop    examples.desktop  Pictures  Templates
bin            Documents  lib               Public    Videos
Customization  Downloads  Music             source
[06/27/19]seed@VM:~$ mkdir lab3
[06/27/19]seed@VM:~$ cd lab3
[06/27/19]seed@VM:~/lab3$ ls
[06/27/19]seed@VM:~/lab3$ echo -n "123456789a" > f2.txt
[06/27/19]seed@VM:~/lab3$ echo -n "12345" > f1.txt
[06/27/19]seed@VM:~/lab3$
[06/27/19]seed@VM:~/lab3$ echo -n "123456789abcdefg" > f3.txt
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in f1.txt -out task4f1_CB
C.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in f2.txt -out task4f2_CB
```

CBC has padding. Padded to 16 bytes when the file size was 5 and 10 bytes.
When the file was 16 bytes, it added an additional block for a total of 32 bytes



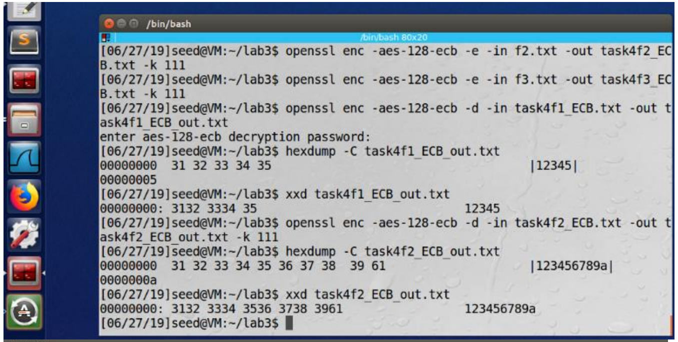
```
hexdump: task4f1_CBC_out.txt.txt: No such file or directory
[06/27/19]seed@VM:~/lab3$ ls
f1.txt  f3.txt               task4f1_CBC.txt  task4f3_CBC.txt
f2.txt  task4f1_CBC_out.txt  task4f2_CBC.txt
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f1_CBC_out.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b 0b  |12345...........|
00000010
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -d -nopad -in task4f2_CBC.txt
 -out task4f2_CBC_out.txt
enter aes-128-cbc decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f1_CBC_out.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b 0b  |12345...........|
00000010
[06/27/19]seed@VM:~/lab3$ xxd task4f1_CBC_out.txt
00000000: 3132 3334 350b 0b0b 0b0b 0b0b 0b0b 0b0b  12345...........
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f2_CBC_out.txt
00000000  31 32 33 34 35 36 37 38  39 61 06 06 06 06 06 06  |123456789a......|
00000010
[06/27/19]seed@VM:~/lab3$ xxd task4f2_CBC_out.txt
00000000: 3132 3334 3536 3738 3961 0606 0606 0606  123456789a......
[06/27/19]seed@VM:~/lab3$
```



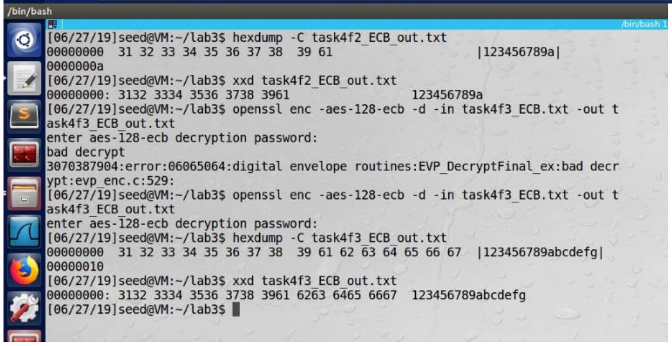
```
enter aes-128-cbc decryption password:
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in f3.txt -out task4f3_CB
C.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -d -nopad -in task4f3_CBC.txt
 -out task4f2_CBC_out.txt
enter aes-128-cbc decryption password:
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in f3.txt -out task4f3_CB
C.txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -d -nopad -in task4f3_CBC.txt
 -out task4f3_CBC_out.txt
enter aes-128-cbc decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f3_CBC_out.txt
00000000  31 32 33 34 35 36 37 38  39 61 62 63 64 65 66 67  |123456789abcdefg|
00000010  10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10  |................|
00000020
[06/27/19]seed@VM:~/lab3$ xxd task4f3_CBC_out.txt
00000000: 3132 3334 3536 3738 3961 6263 6465 6667  123456789abcdefg
00000010: 1010 1010 1010 1010 1010 1010 1010 1010  ................
[06/27/19]seed@VM:~/lab3$
```

ECB

The 5- and 10-byte files were 32 bytes after encryption. The 16-byte file was 48 bytes. There does appear to be padding.



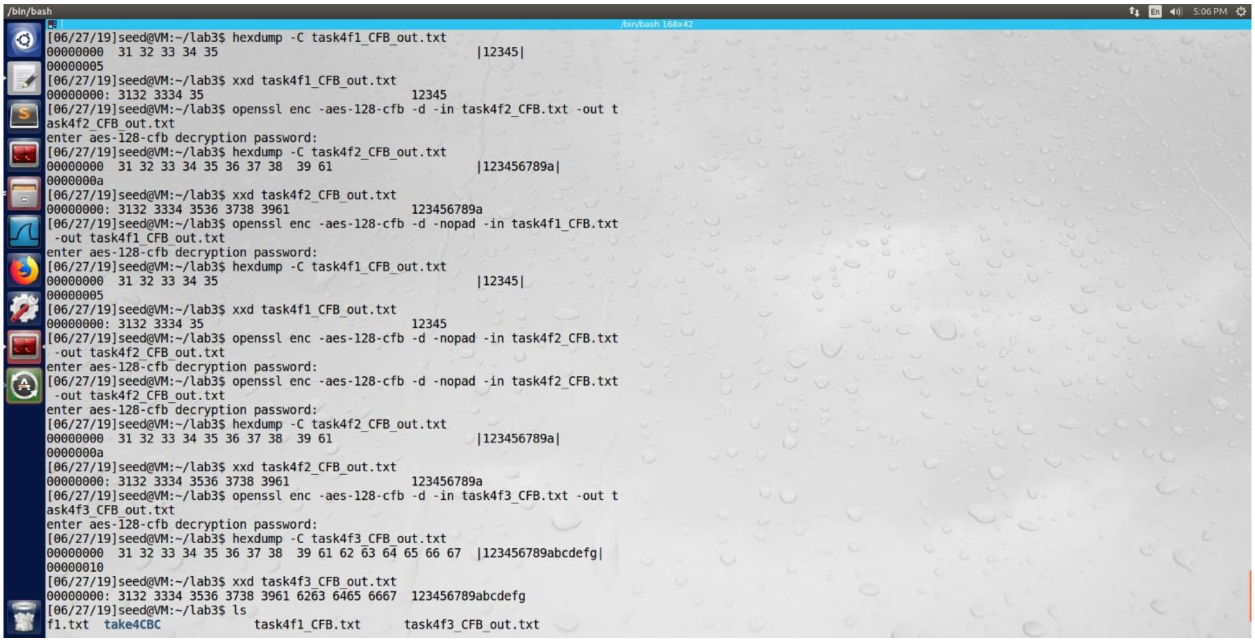
```
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ecb -e -in f2.txt -out task4f2_EC
B.txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ecb -e -in f3.txt -out task4f3_EC
B.txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ecb -d -in task4f1_ECB.txt -out t
ask4f1_ECB_out.txt
enter aes-128-ecb decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f1_ECB_out.txt
00000000  31 32 33 34 35                                    |12345|
00000005
[06/27/19]seed@VM:~/lab3$ xxd task4f1_ECB_out.txt
00000000: 3132 3334 35                             12345
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ecb -d -in task4f2_ECB.txt -out t
ask4f2_ECB_out.txt -k 111
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f2_ECB_out.txt
00000000  31 32 33 34 35 36 37 38  39 61                    |123456789a|
0000000a
[06/27/19]seed@VM:~/lab3$ xxd task4f2_ECB_out.txt
00000000: 3132 3334 3536 3738 3961                 123456789a
[06/27/19]seed@VM:~/lab3$
```



```
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f2_ECB_out.txt
00000000  31 32 33 34 35 36 37 38  39 61                    |123456789a|
0000000a
[06/27/19]seed@VM:~/lab3$ xxd task4f2_ECB_out.txt
00000000: 3132 3334 3536 3738 3961                 123456789a
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ecb -d -in task4f3_ECB.txt -out t
ask4f3_ECB_out.txt
enter aes-128-ecb decryption password:
bad decrypt
3070387904:error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decr
ypt:evp_enc.c:529:
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ecb -d -in task4f3_ECB.txt -out t
ask4f3_ECB_out.txt
enter aes-128-ecb decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f3_ECB_out.txt
00000000  31 32 33 34 35 36 37 38  39 61 62 63 64 65 66 67  |123456789abcdefg|
00000010
[06/27/19]seed@VM:~/lab3$ xxd task4f3_ECB_out.txt
00000000: 3132 3334 3536 3738 3961 6263 6465 6667  123456789abcdefg
[06/27/19]seed@VM:~/lab3$
```

CFB

The 5, 10, and 16 bytes files were encrypted into 21, 26, and 32 byte sized files, respectively. There is no padding. I believe since CFB turns into a stream cipher there is no need for padding.



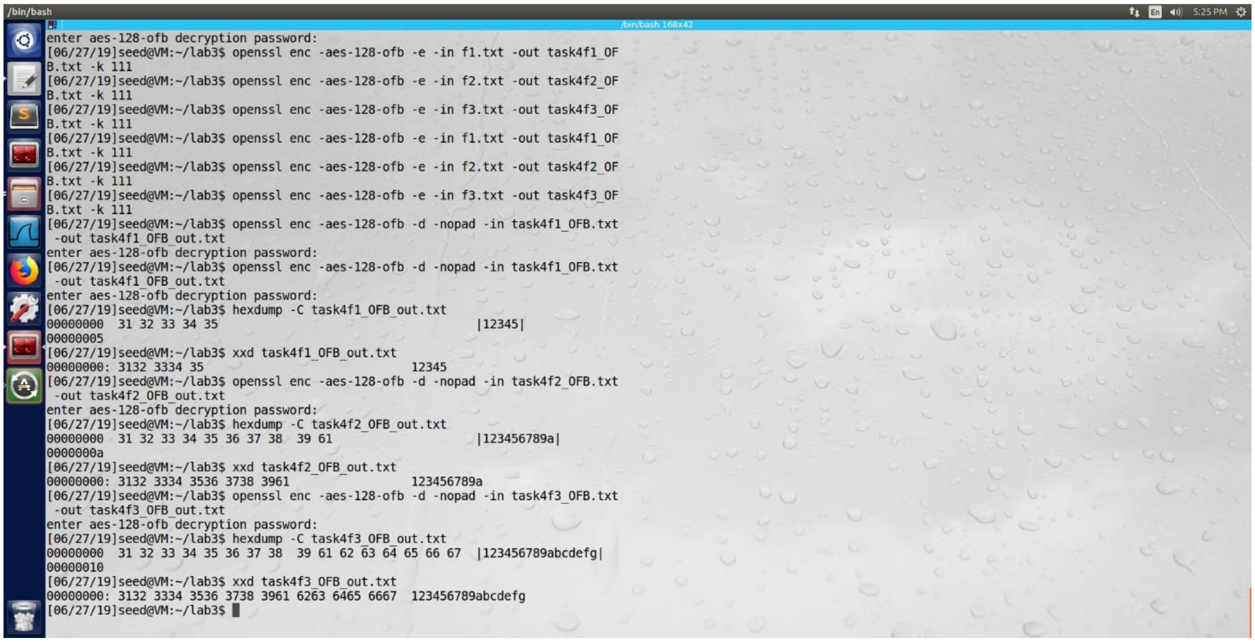
```
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f1_CFB_out.txt
00000000  31 32 33 34 35                                    |12345|
00000005
[06/27/19]seed@VM:~/lab3$ xxd task4f1_CFB_out.txt
00000000: 3132 3334 35                             12345
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cfb -d -in task4f2_CFB.txt -out t
ask4f2_CFB_out.txt
enter aes-128-cfb decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f2_CFB_out.txt
00000000  31 32 33 34 35 36 37 38  39 61                    |123456789a|
0000000a
[06/27/19]seed@VM:~/lab3$ xxd task4f2_CFB_out.txt
00000000: 3132 3334 3536 3738 3961                 123456789a
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cfb -d -nopad -in task4f1_CFB.txt
 -out task4f1_CFB_out.txt
enter aes-128-cfb decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f1_CFB_out.txt
00000000  31 32 33 34 35                                    |12345|
00000005
[06/27/19]seed@VM:~/lab3$ xxd task4f1_CFB_out.txt
00000000: 3132 3334 35                             12345
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cfb -d -nopad -in task4f2_CFB.txt
 -out task4f2_CFB_out.txt
enter aes-128-cfb decryption password:
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cfb -d -nopad -in task4f2_CFB.txt
 -out task4f2_CFB_out.txt
enter aes-128-cfb decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f2_CFB_out.txt
00000000  31 32 33 34 35 36 37 38  39 61                    |123456789a|
0000000a
[06/27/19]seed@VM:~/lab3$ xxd task4f2_CFB_out.txt
00000000: 3132 3334 3536 3738 3961                 123456789a
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cfb -d -in task4f3_CFB.txt -out t
ask4f3_CFB_out.txt
enter aes-128-cfb decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f3_CFB_out.txt
00000000  31 32 33 34 35 36 37 38  39 61 62 63 64 65 66 67  |123456789abcdefg|
00000010
[06/27/19]seed@VM:~/lab3$ xxd task4f3_CFB_out.txt
00000000: 3132 3334 3536 3738 3961 6263 6465 6667  123456789abcdefg
[06/27/19]seed@VM:~/lab3$ ls
f1.txt  take4CBC            task4f1_CFB.txt      task4f3_CFB_out.txt
```

OFB

The files were encrypted into 21, 26, and 32 bytes respectively. This is the same as CFB and I was worried I made a mistake, but after deleting and re-encrypting
it does appear correct, I think.

There does not appear to be padding. I believe since OFB turns into a stream cipher it does not need padding.

https://crypto.stackexchange.com/questions/2072/are-cfb-and-ofb-really-meant-for-streaming



```
enter aes-128-ofb decryption password:
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -e -in f1.txt -out task4f1_OF
B.txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -e -in f2.txt -out task4f2_OF
B.txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -e -in f3.txt -out task4f3_OF
B.txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -e -in f1.txt -out task4f1_OF
B.txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -e -in f2.txt -out task4f2_OF
B.txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -e -in f3.txt -out task4f3_OF
B.txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -d -nopad -in task4f1_OFB.txt
 -out task4f1_OFB_out.txt
enter aes-128-ofb decryption password:
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -d -nopad -in task4f1_OFB.txt
 -out task4f1_OFB_out.txt
enter aes-128-ofb decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f1_OFB_out.txt
00000000  31 32 33 34 35                                    |12345|
00000005
[06/27/19]seed@VM:~/lab3$ xxd task4f1_OFB_out.txt
00000000: 3132 3334 35                             12345
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -d -nopad -in task4f2_OFB.txt
 -out task4f2_OFB_out.txt
enter aes-128-ofb decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f2_OFB_out.txt
00000000  31 32 33 34 35 36 37 38  39 61                    |123456789a|
0000000a
[06/27/19]seed@VM:~/lab3$ xxd task4f2_OFB_out.txt
00000000: 3132 3334 3536 3738 3961                 123456789a
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -d -nopad -in task4f3_OFB.txt
 -out task4f3_OFB_out.txt
enter aes-128-ofb decryption password:
[06/27/19]seed@VM:~/lab3$ hexdump -C task4f3_OFB_out.txt
00000000  31 32 33 34 35 36 37 38  39 61 62 63 64 65 66 67  |123456789abcdefg|
00000010
[06/27/19]seed@VM:~/lab3$ xxd task4f3_OFB_out.txt
00000000: 3132 3334 3536 3738 3961 6263 6465 6667  123456789abcdefg
[06/27/19]seed@VM:~/lab3$ 
```

2.5 Task 5:

I'm not really sure which if any of the encryption methods will actually allow for the recovery of encrypted files.

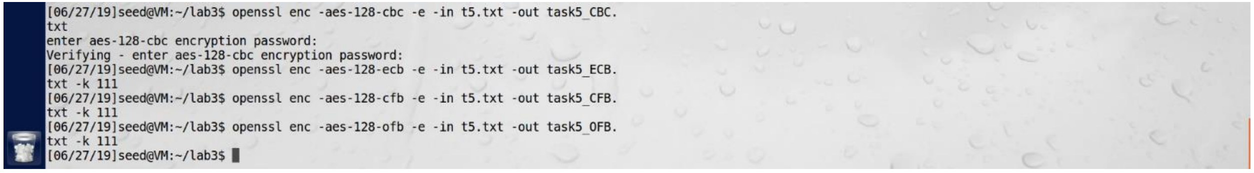
I'm rationalizing that padding will have an impact. Looking at the ways the different encryption methods utilize padding I will guess that:

CDC will be able to recover. I think this is a new method, and I'm guessing it can be recovered because it is newer (and better?). Although,

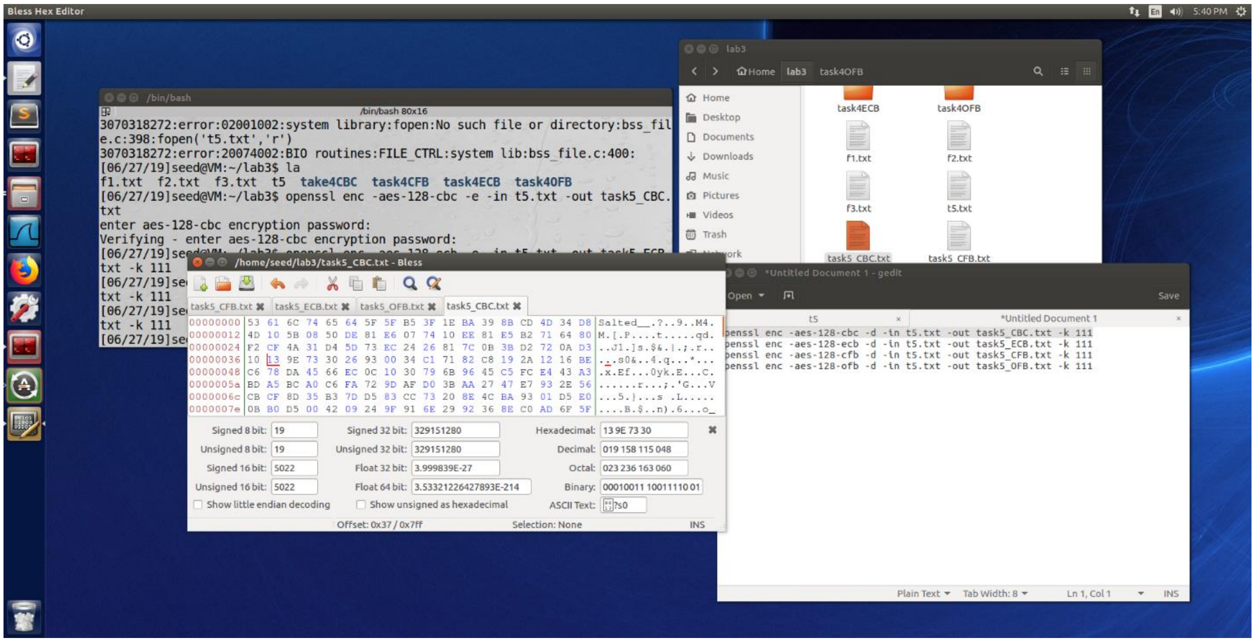
from what I was reading earlier it encrypts based on previously encrypted blocks, so maybe this will attribute to it inability to recover.

ECB I think will be recoverable. The padding had a pattern to it, and I think this will allow for the decryption to make correct itself.

CFB and OFB I'm guessing they will not be able to recover.



```
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t5.txt -out task5_CBC.
txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ecb -e -in t5.txt -out task5_ECB.
txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cfb -e -in t5.txt -out task5_CFB.
txt -k 111
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -e -in t5.txt -out task5_OFB.
txt -k 111
[06/27/19]seed@VM:~/lab3$ 
```



```
3070318272:error:02001002:system library:fopen:No such file or directory:bss_fil
e.c:398:fopen('t5.txt','r')
3070318272:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:400:
[06/27/19]seed@VM:~/lab3$ la
f1.txt  f2.txt  f3.txt  t5  take4CBC  task4CFB  task4ECB  task4OFB
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t5.txt -out task5_CBC.
txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[06/27/19]se
txt -k 111
[06/27/19]se
txt -k 111
[06/27/19]se
txt -k 111
[06/27/19]se
```

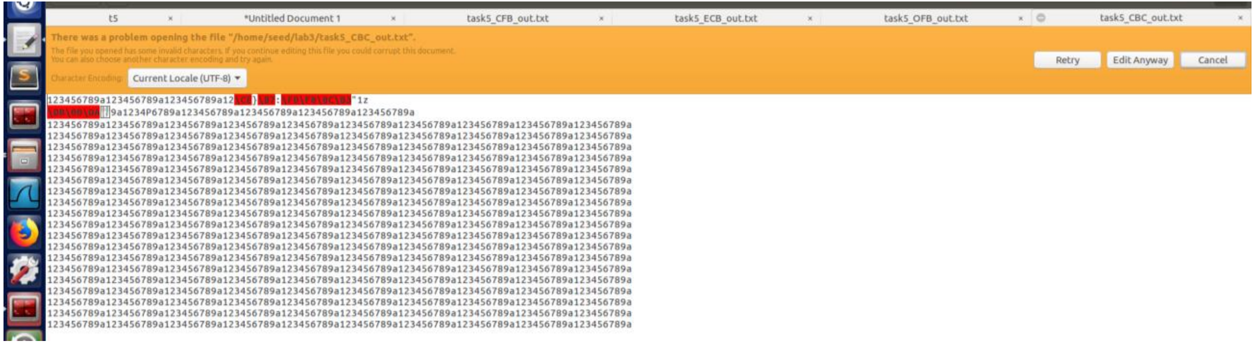
```
00000000 53 61 6C 74 65 64 5F 5F B5 3F 1E BA 39 8B CD 4D 34 D8 Salted__.?..9..M4.
00000012 4D 10 5B 08 50 DE 81 E6 07 74 10 EE 81 E5 B2 71 64 80 M.[.P....t.....qd.
00000024 F2 CF 4A 31 D4 5D 73 EC 24 26 81 7C 0B 3B D2 72 0A D3 ..J1.]s.$&.|.;.r..
00000036 10 13 9E 73 30 26 93 00 34 C1 71 82 C8 19 2A 12 16 BE ...s0&..4.q...*...
00000048 C6 78 DA 45 66 EC 0C 10 30 79 6B 96 45 C5 FC E4 43 A3 .x.Ef...0yk.E...C.
0000005a BD A5 BC A0 C6 FA 72 9D AF D0 3B AA 27 47 E7 93 2E 56 ......r...;.'G...V
0000006c CB CF 8D 35 B3 7D D5 83 CC 73 20 8E 4C BA 93 01 D5 E0 ...5.}...s .L.....
0000007e 0B B0 D5 00 42 09 24 9F 91 6E 29 92 36 8E C0 AD 6F 5F ....B.$..n).6...o_
```

Signed 8 bit:	19	Signed 32 bit:	329151280	Hexadecimal:	13 9E 73 30
Unsigned 8 bit:	19	Unsigned 32 bit:	329151280	Decimal:	019 158 115 048
Signed 16 bit:	5022	Float 32 bit:	3.999839E-27	Octal:	023 236 163 060
Unsigned 16 bit:	5022	Float 64 bit:	3.53321226427893E-214	Binary:	00010011 10011110 01
□ Show little endian decoding		□ Show unsigned as hexadecimal		ASCII Text:	?s0

Offset: 0x37 / 0x7ff   Selection: None   INS

```
penssl enc -aes-128-cbc -d -in t5.txt -out task5_CBC.txt -k 111
penssl enc -aes-128-ecb -d -in t5.txt -out task5_ECB.txt -k 111
penssl enc -aes-128-cfb -d -in t5.txt -out task5_CFB.txt -k 111
penssl enc -aes-128-ofb -d -in t5.txt -out task5_OFB.txt -k 111
```

CBC - Was not able to recover. Upon opening the decrypted file, system sent an error, but let me open anyways. Significant portion of file corrupted.



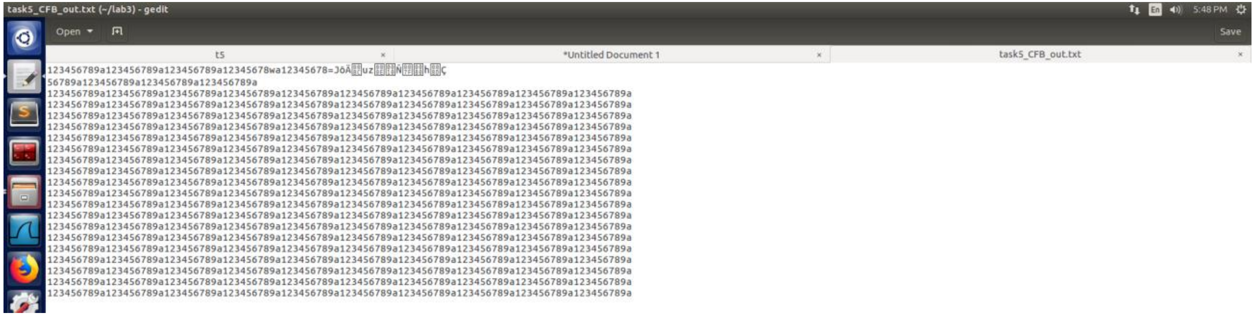
There was a problem opening the file "/home/seed/lab3/task5_CBC_out.txt".

The file you opened has some invalid characters. If you continue editing this file you could corrupt this document.
You can also choose another character encoding and try again.

Character Encoding: Current Locale (UTF-8)

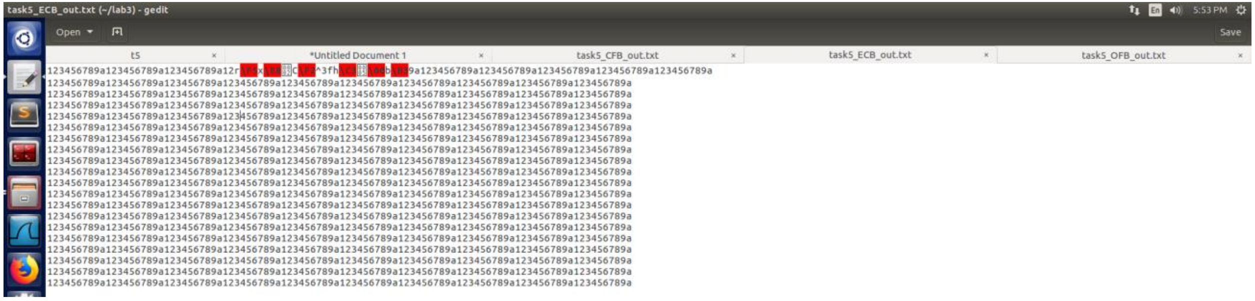
```
123456789a123456789a123456789a12\C6}\B7:\F0\F8\0C\B3"1z
\DB\0D\0A9a1234P6789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
```

CFB - After making the change to 55th byte and decrypting the file was not able to fully recover. Significant portion of the file was corrupted.



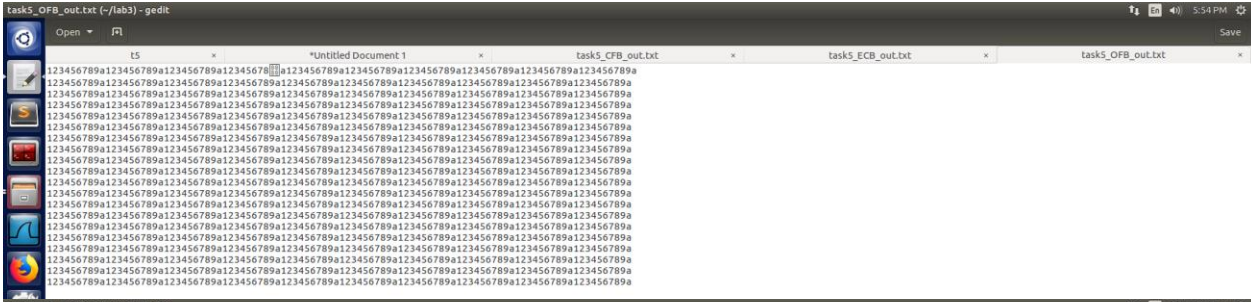
```
123456789a123456789a123456789a12345678wa12345678=JöÄ□uz□□Ñ□□h□Ç
56789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
```

ECB - Was not able to recover. Upon opening file, I received an error opening. It gave me the option to proceed anyways, but the file was corrupted. A significant amount of data in the file appears to be corrupted.

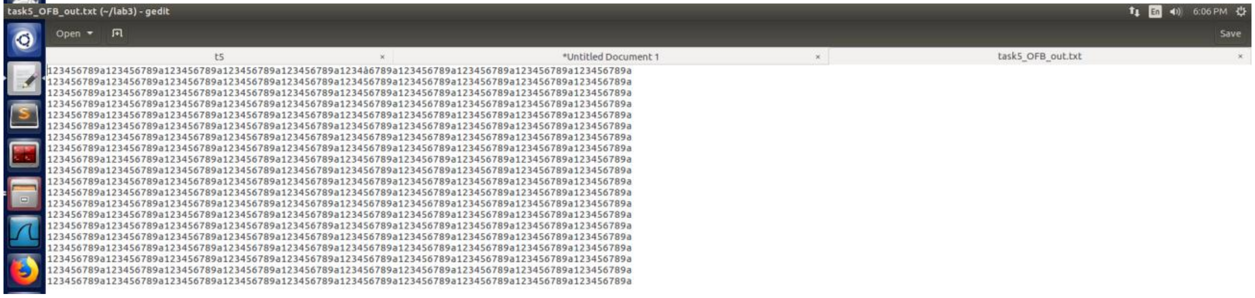


```
123456789a123456789a123456789a12r\F5x\E8□C\F2^3fh\C3□\0db\B39a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
```

OFB - Was not able to fully recover, but only ONE character appears to be corrupted. The rest of the file appears to be fine. (I ran it twice, that is why there are 2 pictures.)



```
123456789a123456789a123456789a12345678□a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
```



```
123456789a123456789a123456789a123456789a123456789a1234å6789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a123456789a
```

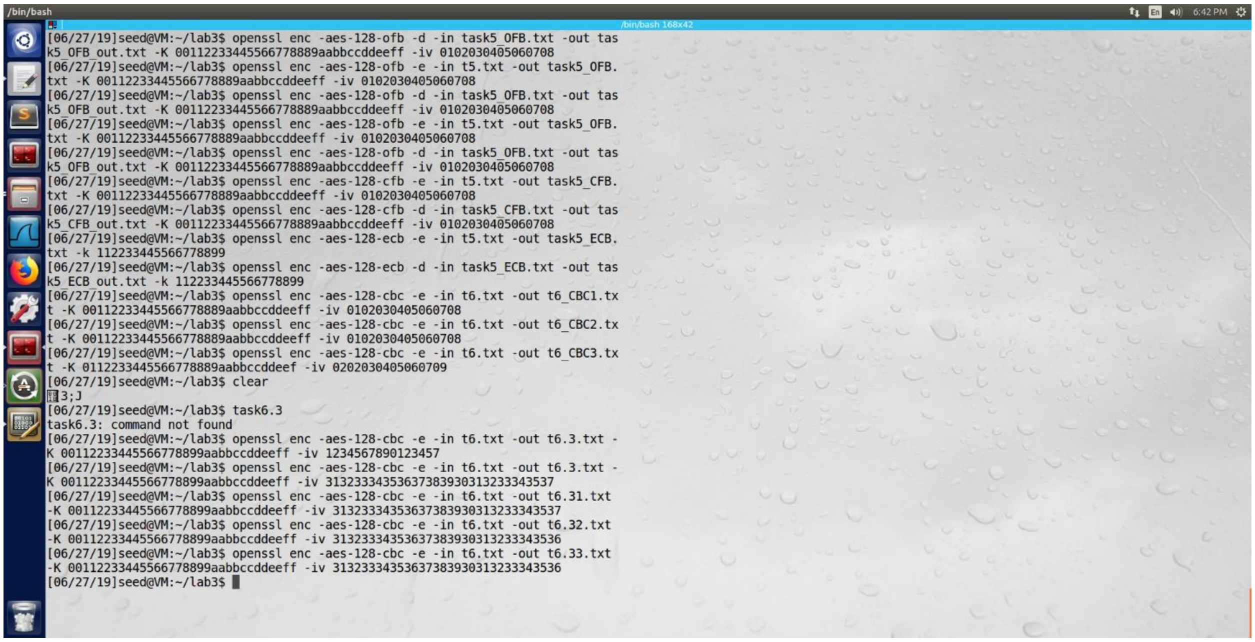
I was very wrong in my initial assumption about which encryptions would recover and vice versa.

2.6 Task 6:

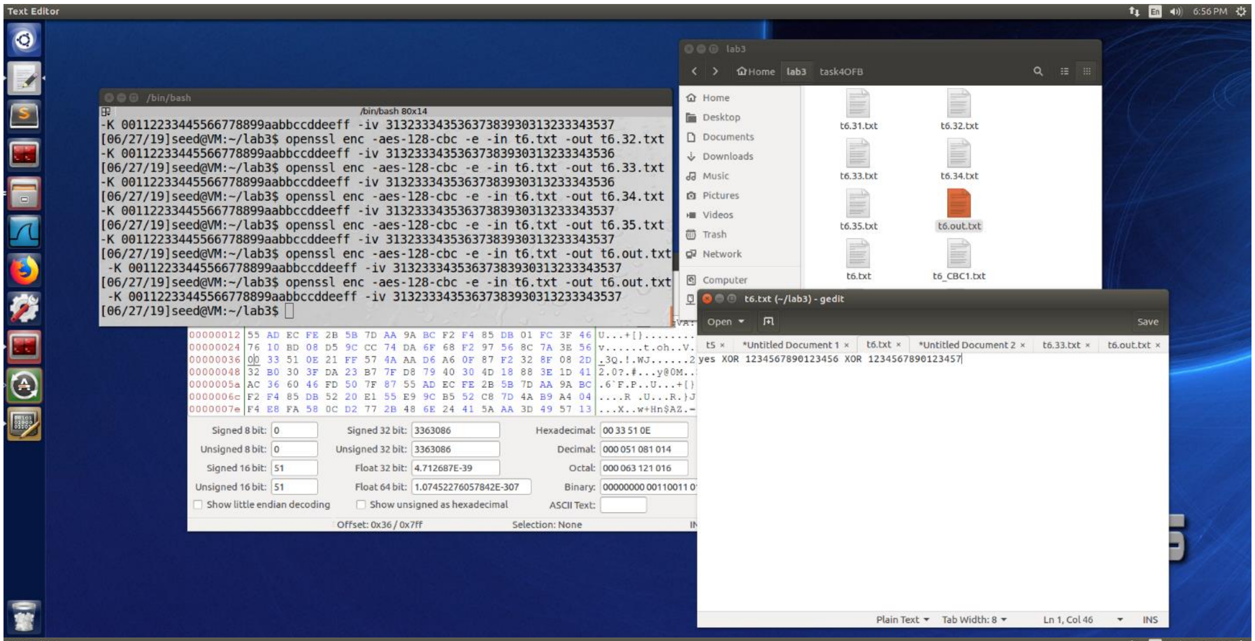
I wasn't able to figure this out. I spent a good amount of time reading about the chosen plaintext attack and about the initialization vector and I couldn't reproduce the results described in the lab.

I understood the idea of using a plaintext with A XOR IVA XOR IVN where A is either "yes" or "no", IVA is the initialization vector used to create A, and IVN is the initialization vector used next (ie. predictable IV). I didn't understand what the XOR was supposed to be in plain text. I tried a lot of different combinations such as "XOR", "^", not including anything, separating by blocks of 16 bytes, using newlines, maybe more, but I couldn't get a cipher text the resembled C1 cipher text.

After some time, I started to think something was wrong. I tried using the key, the plaintext and IVA to try to recreate the cipher text given (C1). This should work, correct?
I'm given the key, the IV, and I know the plaintext is either yes or no. I should be able to recreate the ciphertext wit this. I tried this. I encrypted both "yes" and "no" using AES 128-bit CDC using the provided key and IV (in hex) and I couldn't even get the cipher text (C1) to reproduce.



```
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -d -in task5_OFB.txt -out tas
k5_OFB_out.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -e -in t5.txt -out task5_OFB.
txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -d -in task5_OFB.txt -out tas
k5_OFB_out.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -e -in t5.txt -out task5_OFB.
txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ofb -d -in task5_OFB.txt -out tas
k5_OFB_out.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cfb -e -in t5.txt -out task5_CFB.
txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cfb -d -in task5_CFB.txt -out tas
k5_CFB_out.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ecb -e -in t5.txt -out task5_ECB.
txt -k 112233445566778899
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-ecb -d -in task5_ECB.txt -out tas
k5_ECB_out.txt -k 112233445566778899
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6_CBC1.tx
t -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6_CBC2.tx
t -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6_CBC3.tx
t -K 0112233445566778889aabbccddeef -iv 0202030405060709
[06/27/19]seed@VM:~/lab3$ clear
3;J
[06/27/19]seed@VM:~/lab3$ task6.3
task6.3: command not found
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.3.txt -
K 00112233445566778899aabbccddeeff -iv 1234567890123457
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.3.txt -
K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.31.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.32.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.33.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ 
```

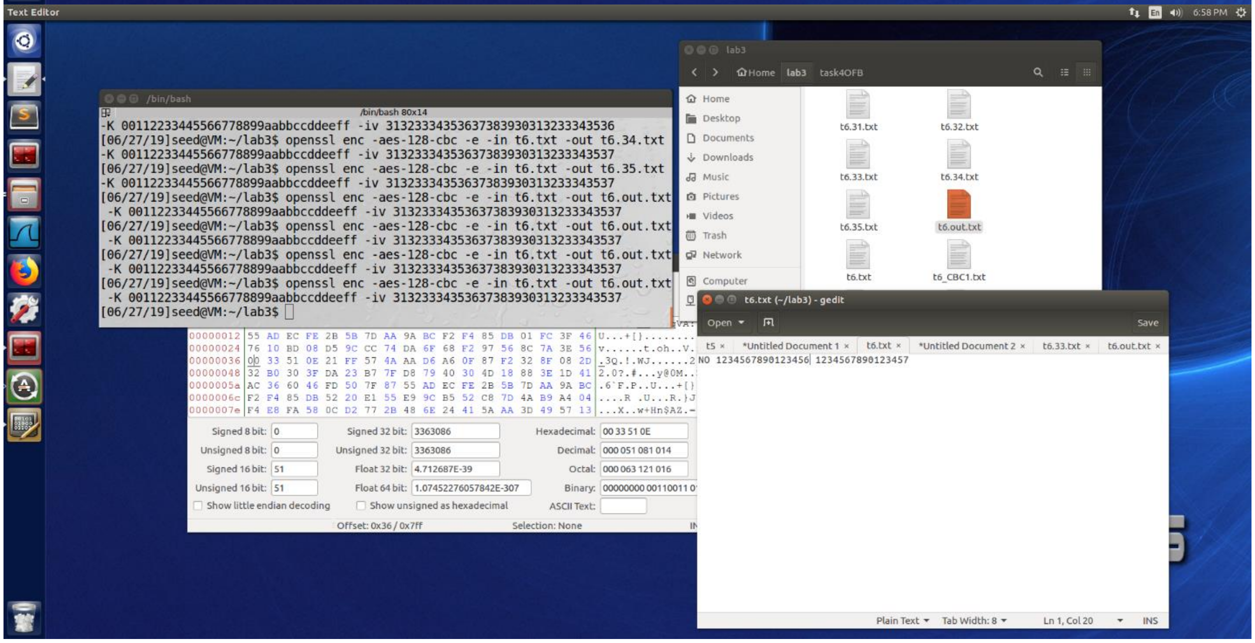


```
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.32.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.33.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.34.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.35.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ 
```

```
00000012 55 AD EC FE 2B 5B 7D AA 9A BC F2 F4 85 DB 01 FC 3F 46 U...+[}........
00000024 76 10 BD 08 D5 9C CC 74 DA 6F 68 F2 97 56 8C 7A 3E 56 v......t.oh..V.
00000036 00 33 51 0E 21 FF 57 4A AA D6 A6 0F 87 F2 32 8F 08 2D .3Q.!.WJ......2
00000048 32 B0 30 3F DA 23 B7 7F D8 79 40 30 4D 18 88 3E 1D 41 2.0?.#...y@0M..
0000005a AC 36 60 46 FD 50 7F 87 55 AD EC FE 2B 5B 7D AA 9A BC .6`F.P..U...+[}
0000006c F2 F4 85 DB 52 20 E1 55 E9 9C B5 52 C8 7D 4A B9 A4 04 ....R .U...R.}J
0000007e F4 E8 FA 58 0C D2 77 2B 48 6E 24 41 5A AA 3D 49 57 13 ...X..w+Hn$AZ.=
```

Signed 8 bit: 0 Signed 32 bit: 3363086 Hexadecimal: 00 33 51 0E
Unsigned 8 bit: 0 Unsigned 32 bit: 3363086 Decimal: 000 051 081 014
Signed 16 bit: 51 Float 32 bit: 4.712687E-39 Octal: 000 063 121 016
Unsigned 16 bit: 51 Float 64 bit: 1.07452276057842E-307 Binary: 00000000 00110011 0
□ Show little endian decoding □ Show unsigned as hexadecimal ASCII Text:
Offset: 0x36 / 0x7ff Selection: None

```
yes XOR 1234567890123456 XOR 1234567890123457
```

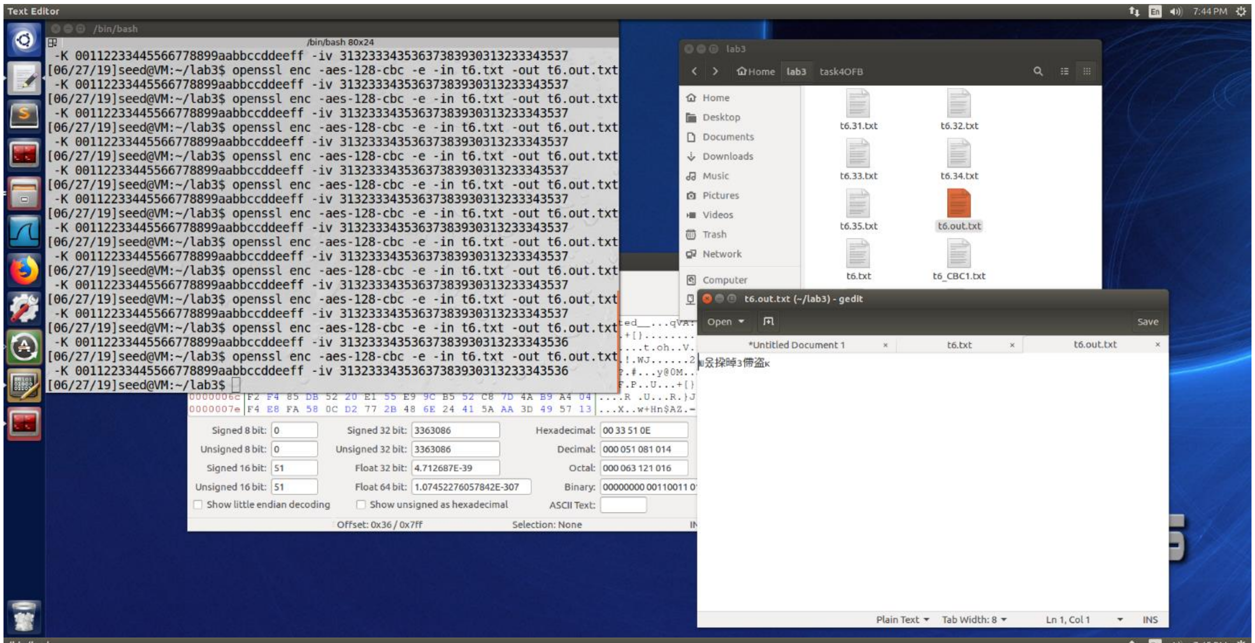


```
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.34.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.35.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ 
```

```
00000012 55 AD EC FE 2B 5B 7D AA 9A BC F2 F4 85 DB 01 FC 3F 46 U...+[}........
00000024 76 10 BD 08 D5 9C CC 74 DA 6F 68 F2 97 56 8C 7A 3E 56 v......t.oh..V.
00000036 00 33 51 0E 21 FF 57 4A AA D6 A6 0F 87 F2 32 8F 08 2D .3Q.!.WJ......2
00000048 32 B0 30 3F DA 23 B7 7F D8 79 40 30 4D 18 88 3E 1D 41 2.0?.#...y@0M..
0000005a AC 36 60 46 FD 50 7F 87 55 AD EC FE 2B 5B 7D AA 9A BC .6`F.P..U...+[}
0000006c F2 F4 85 DB 52 20 E1 55 E9 9C B5 52 C8 7D 4A B9 A4 04 ....R .U...R.}J
0000007e F4 E8 FA 58 0C D2 77 2B 48 6E 24 41 5A AA 3D 49 57 13 ...X..w+Hn$AZ.=
```

Signed 8 bit: 0 Signed 32 bit: 3363086 Hexadecimal: 00 33 51 0E
Unsigned 8 bit: 0 Unsigned 32 bit: 3363086 Decimal: 000 051 081 014
Signed 16 bit: 51 Float 32 bit: 4.712687E-39 Octal: 000 063 121 016
Unsigned 16 bit: 51 Float 64 bit: 1.07452276057842E-307 Binary: 00000000 00110011 0
□ Show little endian decoding □ Show unsigned as hexadecimal ASCII Text:
Offset: 0x36 / 0x7ff Selection: None

```
NO 1234567890123456 1234567890123457
```

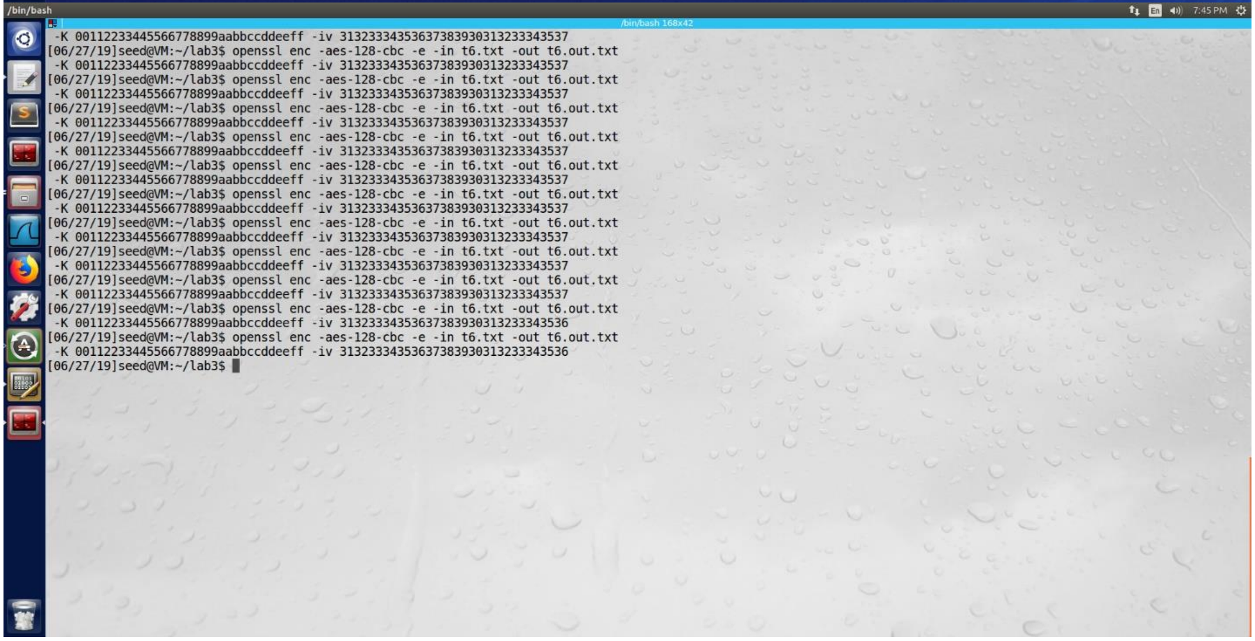


```
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ 
```

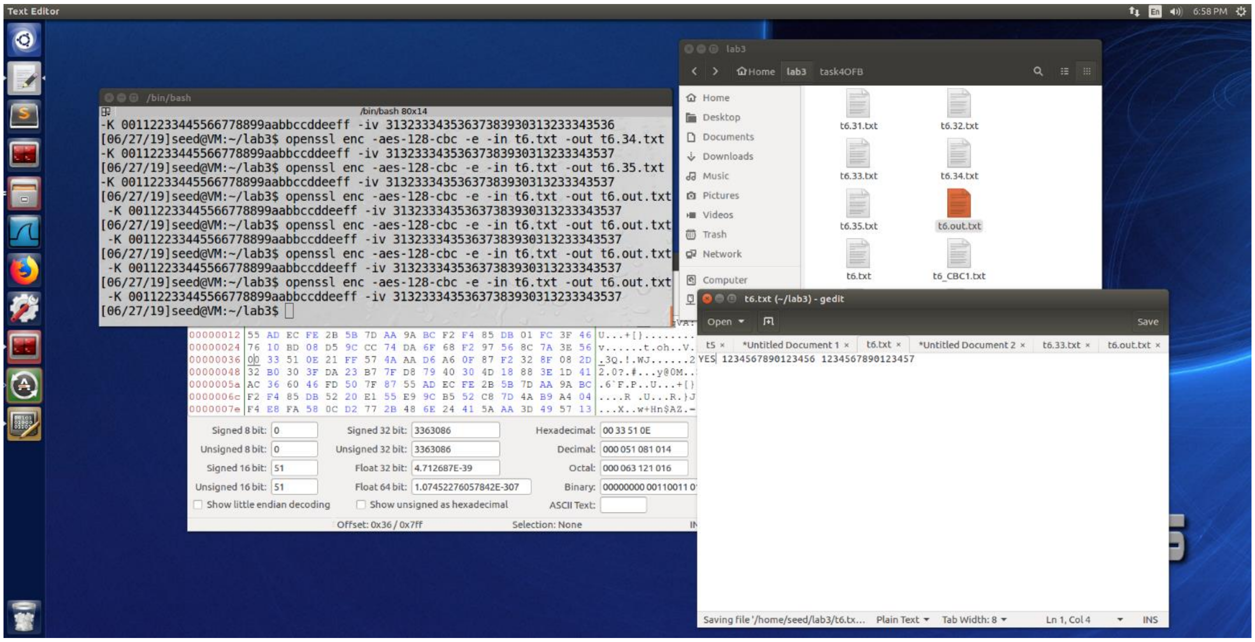
```
ted__...qVA:
.+[}........
....t.oh..V.
.!.WJ......2
?.#...y@0M..
F.P..U...+[}
0000006c F2 F4 85 DB 52 20 E1 55 E9 9C B5 52 C8 7D 4A B9 A4 04 ....R .U...R.}J
0000007e F4 E8 FA 58 0C D2 77 2B 48 6E 24 41 5A AA 3D 49 57 13 ...X..w+Hn$AZ.=
```

Signed 8 bit: 0
Signed 32 bit: 3363086
Hexadecimal: 00 33 51 0E
Unsigned 8 bit: 0
Unsigned 32 bit: 3363086
Decimal: 000 051 081 014
Signed 16 bit: 51
Float 32 bit: 4.712687E-39
Octal: 000 063 121 016
Unsigned 16 bit: 51
Float 64 bit: 1.07452276057842E-307
Binary: 00000000 00110011 0
□ Show little endian decoding
□ Show unsigned as hexadecimal
ASCII Text:
Offset: 0x36 / 0x7ff
Selection: None

炁搡暐3偋盗к



```
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ 
```



```
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343536
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.34.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.35.txt
-K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ openssl enc -aes-128-cbc -e -in t6.txt -out t6.out.txt
 -K 00112233445566778899aabbccddeeff -iv 31323334353637383930313233343537
[06/27/19]seed@VM:~/lab3$ 
```

```
00000012 55 AD EC FE 2B 5B 7D AA 9A BC F2 F4 85 DB 01 FC 3F 46 U...+[}........
00000024 76 10 BD 08 D5 9C CC 74 DA 6F 68 F2 97 56 8C 7A 3E 56 v......t.oh..V.
00000036 00 33 51 0E 21 FF 57 4A AA D6 A6 0F 87 F2 32 8F 08 2D .3Q.!.WJ......2
00000048 32 B0 30 3F DA 23 B7 7F D8 79 40 30 4D 18 88 3E 1D 41 2.0?.#...y@0M..
0000005a AC 36 60 46 FD 50 7F 87 55 AD EC FE 2B 5B 7D AA 9A BC .6`F.P..U...+[}
0000006c F2 F4 85 DB 52 20 E1 55 E9 9C B5 52 C8 7D 4A B9 A4 04 ....R .U...R.}J
0000007e F4 E8 FA 58 0C D2 77 2B 48 6E 24 41 5A AA 3D 49 57 13 ...X..w+Hn$AZ.=
```

Signed 8 bit:	0	Signed 32 bit:	3363086	Hexadecimal:	00 33 51 0E
Unsigned 8 bit:	0	Unsigned 32 bit:	3363086	Decimal:	000 051 081 014
Signed 16 bit:	51	Float 32 bit:	4.712687E-39	Octal:	000 063 121 016
Unsigned 16 bit:	51	Float 64 bit:	1.07452276057842E-307	Binary:	00000000 00110011 0
□ Show little endian decoding		□ Show unsigned as hexadecimal		ASCII Text:	

Offset: 0x36 / 0x7ff   Selection: None

```
YES 1234567890123456 1234567890123457
```

Saving file '/home/seed/lab3/t6.tx...   Plain Text   Tab Width: 8   Ln 1, Col 4   INS