

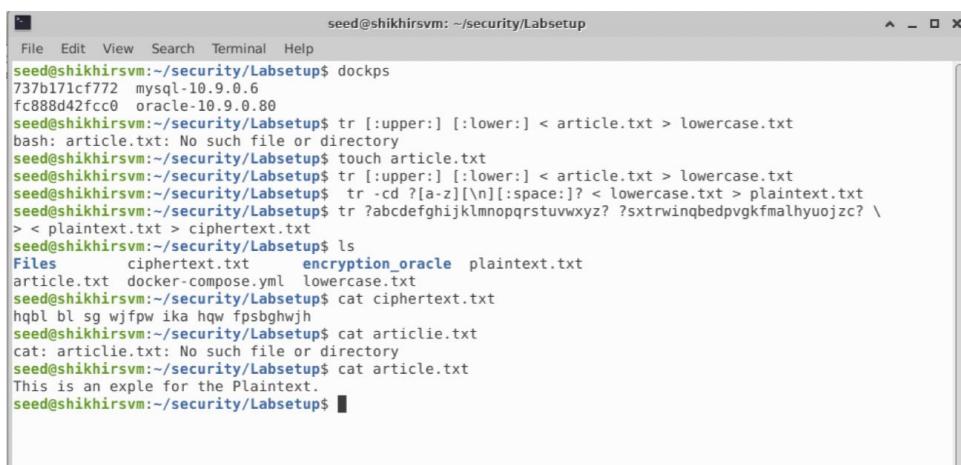
# CS 201P Project #7— Secret-Key Encryption Lab

**Name:** Shikhir Goel

**UCI ID:** [shikhrg@uci.edu](mailto:shikhrg@uci.edu)

**Computing/Cloud Platform Chosen:** Google Cloud platform

## Task 1: Frequency Analysis



The screenshot shows a terminal window titled "seed@shikhirsrvm: ~/security/Labsetup". The terminal displays a series of shell commands used for frequency analysis. It starts with "dockps" to start a Docker container, followed by tr commands to convert uppercase to lowercase and then to remove punctuation. A file named "article.txt" is mentioned but not found. The user then creates an "article.txt" file and processes it again. Finally, they run "cat ciphertext.txt" and "cat article.txt" to compare them, noting that the ciphertext is a shuffled version of the plaintext.

```
seed@shikhirsrvm:~/security/Labsetup$ dockps  
737b171cf772 mysql-10.9.0.6  
fc888d42fcc0 oracle-10.9.0.80  
seed@shikhirsrvm:~/security/Labsetup$ tr [:upper:] [:lower:] < article.txt > lowercase.txt  
bash: article.txt: No such file or directory  
seed@shikhirsrvm:~/security/Labsetup$ touch article.txt  
seed@shikhirsrvm:~/security/Labsetup$ tr [:upper:] [:lower:] < article.txt > lowercase.txt  
seed@shikhirsrvm:~/security/Labsetup$ tr -cd ?[a-z][\n][:space:]? < lowercase.txt > plaintext.txt  
seed@shikhirsrvm:~/security/Labsetup$ tr ?abcdefghijklmnopqrstuvwxyz? ?sxtrwinqbcdpvgkfmalyuojzc? \  
> < plaintext.txt > ciphertext.txt  
seed@shikhirsrvm:~/security/Labsetup$ ls  
Files      ciphertext.txt      encryption_oracle     plaintext.txt  
article.txt  docker-compose.yml  lowercase.txt  
seed@shikhirsrvm:~/security/Labsetup$ cat ciphertext.txt  
hqlb bl sg wifp ika hqo fspbghwjh  
seed@shikhirsrvm:~/security/Labsetup$ cat articlie.txt  
cat: articlie.txt: No such file or directory  
seed@shikhirsrvm:~/security/Labsetup$ cat article.txt  
This is an exple for the Plaintext.  
seed@shikhirsrvm:~/security/Labsetup$
```

ytn xqavhq yzhu xu qzupvd ltmat qnncq vxgxy harty vbynh ytmq ixur qyhnurn  
vlvhpq yhme ytn gvrnnh bnniq imsn v uxuvrnuvhmvu yxx  
ytn vlvhpq hvan lvq qxxanupnp gd ytn pncmqa xb tvhfnd lmmuqymmu vy myq xzyqny  
vup ytn veehnru mceikgnku xb tmq bmlc axcevud vy ytn nup vup my lvq qtvenp gd  
ytn ncnhrnua xb cnyxx ymcnq ze givasrxlu eximymaq vhacavupd vayfmqec vup  
v uvymxuvi axutnhqgvnxu vq ghmnbd vup cvp vq v bnfnh phnvc vxgxy ltnytnh ytnhn  
kzrty yx gn v ehnqmpnuy lmubhnd ytn qnvqxu pmpuy ozqy qnnc nkyhv ixur my lvq  
skyhv ixur gnazvqn ytn xqavhq lnhn cxfnq yx ytn bmhqv lnnsnup mu cvhat yx  
vrxmp axubimaymusr lmyt ytn aixqmur anhncxud xb ytn lmuyhn xidicemaq ytvusq  
ednxuraturvur

xun gmr jznqyaxu qzhhxzupmusr ytmq dnvhq vavpnoc vlvhpq mq txl xh mb ytn  
anhncxud lmi i vpphnqq cnyxx nqenamviid vbynh ytn rxipnu rixgnq ltmat qnaven  
v ozgmiuyv axcmurkzy evhyd bxh ymcnq ze ytn cxfncnuy qenvhtrvpnp gd  
exlnhhbzl txilidixxp lxenu ltx tnieng hvnqhn cmiiimxu xb pxlivhg yx bmrtv qakzvi  
vvhvqgcnuy vhxxup ytn axzuyhd

gnruvimir ytnmh qzeexhy rxipnu rixgnq vyyupnnq qlvytvp ytnognifnq mu givas  
qexhynp iveni emuq vup qxuzupnp xbb vxgxy qnkmgq exlnh mcgvivuang bhxc ytn hnp  
avheny vup ytn qyvrn xu ytn vmb n lvq aviihp xxy vxgxy evd munizmyd vbynh  
nyq bxhcnh vuatxh avyy qvpinh jzmy xuan qta invhump ytvv qtn lvq cvsmur bvh  
lnqq ytvu v cvin axtkqy vup pznmur ytn anhncxud uvylmim exhyvcu yxss v gizuy  
vup qvymqbdmusr pmr vy ytn viievin hxqyhn xb uxcmuvnp pmhnayhq txl axzip  
ytvy gn yxeenp

vq my yzhuq xxy vy invqy mu ynhoq xb ytn xqavhq my ehxvgvid ixuy gn

lxenu mufxitnp mu ymcnq ze qvmp ytvv viytxzrt ytn rixgnq qmrumbmnp ytn  
numymvymfnq ivzust ytnd unfnh muynupnp my yx gn ozqy vu vlvhpq qnvqxu  
avcoevmru xb xun ytvv gnacn vqqxamvyp xuid lmyt hnpavheny waymxuq muqyvnp  
v qexsnglxvnu qvmp ytn rhxxe mq lkhsmur qntmup aixqnp pxxhq vup tvq qmuan  
rcvqgnp cmiiimxu bxh myq inrvi pbnuqng bzup ltmat vbynh ytn rixgnq lvq  
pixxnpn lmyt ytxzqvupq xb pxuvymxu xb xh inqq bhxc enxein mu qxen  
ixzuyhmnq

ix avii yx lnh givaa rxliq inuy xxy mu vpfvuan xb ytn xqavhq ytxzrt ytn  
cxfnconuy lmi i vickqy anhyvuid gn hnbnhnua np gnbxhn vup pznmur ytn anhncxud  
nqenamviid qmuan fxavi cnyxx qzeexhyhgnq imsn vgtind czpp ivzhv pnhu vup  
maxin smpcvu vhn qatnpzinp ehnqnuhnq

vuxylnh bnvyzhn xb ytmq qnvqxu ux xun hnviid suslxl ttx mq rxmurr yx lmu gnqy  
emayzhn vrhrzvpid ytmq tweenuq v ixy xb ytn ymcn muvhrzvpid ytn uvnigymn  
vvhvymfnx uid qnhfnq ytn vlvhpq tden cvatmn qzy xbynu ytn enxein bxhnawqymur  
ytn hvan gxaviinp xqavhxixrmqyq avu cvan xuid pzzavypn rznqng  
ytn lvd ytn vavpnoc yvgzivnq ytn gmr lmuunh pxnqy tnie mu nfndd xytnh  
avynrxhd ytn uxcmunn lmyt ytn cxqy fxynq lmuq gzy mu ytn gnqy emayzhn  
avynrxhd fxynq vhn vgsnp yx imqy ytmah yxe cxfmnp mu ehnbnhnuyvni xhpnh mb v  
cxfmn rnyq cxhn enhanuy xb ytn bmhqeivan fxynq my lmuq ltnu ux

This is our ciphertext provided to us in the labsetup file. Using the frequency analysis method, we will try to decode this ciphertext into plaintext.

## **Bigrams**

Of 2,383,373,483 bigrams scanned:

1. th (92535489, 3.882543%)
2. he (87741289, 3.681391%)
3. in (54433847, 2.283899%)
4. er (51910883, 2.178042%)
5. an (51015163, 2.140460%)
6. re (41694599, 1.749394%)
7. nd (37466077, 1.571977%)
8. on (33802063, 1.418244%)
9. en (32967758, 1.383239%)
10. at (31830493, 1.335523%)
11. ou (30637892, 1.285484%)
12. ed (30406590, 1.275779%)
13. ha (30381856, 1.274742%)
14. to (27877259, 1.169655%)
15. or (27434858, 1.151094%)
16. it (27048699, 1.134891%)
17. is (26452510, 1.109877%)
18. hi (26033632, 1.092302%)
19. es (26033602, 1.092301%)
20. ng (25106109, 1.053385%)

## Trigrams

Of 1,699,542,842 trigrams scanned:

1. the (59623899, 3.508232%)
2. and (27088636, 1.593878%)
3. ing (19494469, 1.147042%)
4. her (13977786, 0.822444%)
5. hat (11059185, 0.650715%)
6. his (10141992, 0.596748%)
7. tha (10088372, 0.593593%)
8. ere (9527535, 0.560594%)
9. for (9438784, 0.555372%)
10. ent (9020688, 0.530771%)
11. ion (8607405, 0.506454%)
12. ter (7836576, 0.461099%)
13. was (7826182, 0.460487%)
14. you (7430619, 0.437213%)
15. ith (7329285, 0.431250%)
16. ver (7320472, 0.430732%)
17. all (7184955, 0.422758%)
18. wit (6752112, 0.397290%)
19. thi (6709729, 0.394796%)
20. tio (6425262, 0.378058%)

We look at the n-gram frequencies of most commonly occurring 2-gram and 3-gram words in the English language.

```

seed@shikhirsrvm:~/security/Labsetup$ cd Files
seed@shikhirsrvm:~/security/Labsetup/Files$ ./freq.py
-----
1-gram (top 20):
n: 488
y: 373
v: 348
x: 291
u: 280
q: 276
m: 264
h: 235
t: 183
i: 166
p: 156
a: 116
c: 104
z: 95
l: 90
g: 83
b: 83
r: 82
e: 76
d: 59
-----
2-gram (top 20):
yt: 115
tn: 89
mu: 74
nh: 58
vh: 57
hn: 57
vu: 56
nq: 53
xu: 52
up: 46
xh: 45
yn: 44
np: 44
vy: 44
nu: 42
qy: 39

```

Using the freq.py file we can see the frequencies of the most commonly occurring n-grams in our ciphertext. Using the frequency analysis and the other links provided in the document, we map the ciphertext alphabets to plain text alphabets.

```

seed@shikhirsrvm:~/security/Labsetup$ ls
Files      docker-compose.yml  file.enc      fileBlowfish.enc  lfcb.bin   lfecbl.txt  lowercase.txt
article.txt  encr.bin        fileAES128.enc  fileaes128cfb.enc  lfcb1.txt  lfecb1.txt  mysql_data
ciphertext.txt encryption_oracle  fileAES256.enc  largefile.txt    lfecb.bin   lfecb1.txt  plaintext.txt
seed@shikhirsrvm:~/security/Labsetup$ tr 'ntyhquvmxbpzfrceialdgoswjk' 'EHTRSNAlOFGDUVGMPCLCWBJKMQX' < ciphertext.txt > out.txt
tr: warning: character map not found in table
seed@shikhirsrvm:~/security/Labsetup$ ls
RSFW FW KB MQVDM LXC RSM VDKFBMRQ
seed@shikhirsrvm:~/security/Labsetup$ cat out.txt
RSFW FW KB MQVDM LXC RSM VDKFBMRQ
seed@shikhirsrvm:~/security/Labsetup$ tr "ntyhquvmxbpzfrceialdgoswjk" "EHTRSNAlOFGDUVGMPCLCWBJKMQX" < ciphertext.txt > out1.txt
tr: warning: character map not found in table
seed@shikhirsrvm:~/security/Labsetup$ ls
RSFW FW KB MQVDM LXC RSM VDKFBMRQ
seed@shikhirsrvm:~/security/Labsetup$ cat out1.txt
RSFW FW KB MQVDM LXC RSM VDKFBMRQ
seed@shikhirsrvm:~/security/Labsetup$ tr "ntyhquvmxbpzfrceialdgoswjk" "EHTRSNAlOFGDUVGMPCLCWBJKMQX" < ciphertext.txt > out1.txt

```

Using the translate function we replace the lowercase ciphertext with capital plaintext and the final output is shown below.

I have used the following command:

```
tr "ntyhquvmxbpzfrceialdgoswjk" "EHTRSNAlOFGDUVGMPCLCWBJKMQX" < ciphertext.txt > out1.txt
```

```
File Edit View Search Terminal Help
article.txt      encryption_oracle fileBlowfish.enc lfcfb1.txt lfobf1.txt out1.txt
ciphertext.txt   file.enc       fileaes128cfb.enc lfecb.bin lowercase.txt plaintext.txt
docker-compose.yml fileAES128.enc largefile.txt lfecb1.txt mysql_data
seed@shikhirsvm:/security/Labsetup$ cat out1.txt
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
DOUGHT 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 SPKESWOMAN 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 ONE SERVES THE AWARDS HYM MACHINE BUT OFTEN THE PEOPPLE FORECASTING
THE RACE SO-CALLED OSCARLOGISTS CAN MAKE INTELLIGENT GUESSES THAT MAY THE ACADEMY TABULATES THE BIG WINNER DOESNT HELP IN EVERY OTHER
CATEGORY THE NOMINEES WITH THE MOST VOTES WINS BUT IN THE BEST PICTURE
CATEGORY VOTERS ARE ASKED TO RANK THE 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
PRIME 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
```

We can see that we have successfully translated/decrypted the ciphertext into plaintext.

## Task 2: Encryption using Different Ciphers and Modes

In this task we encrypt the files using different ciphers and different modes.

```
shikher@shikhsvm: /home/seed/security/Labsetup - Google Chrome
ssh.cloud.google.com/projects/serene-cathode-329819/zones/us-central1-a/instances/shikhsvm?authuser=0&hl=en_US&projectNumber=588356155406&useAdminProxy=true&troubleshoot4005...
Verifying - enter aes-256-cbc encryption password:
Can't open dangle for writing, Permission denied
139798693045568:error:0200100D:system library:fopen:Permission denied:./crypto/b
io/bas file.c:69:fopen("dangle","w")
139798693045568:error:20060002:BIO routines:BIO_new_file:system lib:./crypto/b
io/bas file.c:78:
shikher@shikhsvm:/home/seed/security/Labsetup$ openssl list-cipher-commands
Invalid command 'list-cipher-commands'; type "help" for a list.
shikher@shikhsvm:/home/seed/security/Labsetup$ openssl enc -aes-256-cbc -pass
pass:shikhir -p -in plaintext.txt -out file.enc
Can't open file.enc for writing, Permission denied
140230302373568:error:0200100D:system library:fopen:Permission denied:./crypto/b
io/bas file.c:69:fopen("file.enc",'wb')
140230302373568:error:20060002:BIO routines:BIO_new_file:system lib:./crypto/b
io/bas file.c:78:
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -aes-256-cbc -
pass pass:shikhir -p -in plaintext.txt -out file.enc
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=F2F161C9A62
key=BSA10600E00X75A357D89A7F748393BCEDDBS108BD8E404E2041FFB8C6FB2B3
iv =1D9776FD024B68C766703AC0105A20
shikher@shikhsvm:/home/seed/security/Labsetup$ ls
Files      ciphertext.txt      encryption oracle      lowercase.txt      plaintext.txt
article.txt docker-compose.yml      file.enc      mysql_data
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -aes-256-cbc -
pass pass:shikhir -p -in plaintext.txt -out fileAES256.enc
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=A0DC8CF6EC57989E9
iv =3D201095YAFZFX40006737820861M3K
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -sha-256-cbc -pass pass:shikhir -p -in plaintext.txt -out fileSHA256.enc
enc: Unrecognized flag sha-256-cbc
enc: Use -help for summary.
shikher@shikhsvm:/home/seed/security/Labsetup$ -help
Command '-help' not found, did you mean:
  command 'dhelp' from deb dhelp (0.6.26)
Try: apt install <deb name>
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -aes-128-cbc -pass pass:shikhir -p -in plaintext.txt -out fileAES128.enc
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=F2F161C9A62
key=E625C4AB9552F384E55EC840A75716F
iv =S57DF51CDC48D1509E7F54AC59C4D7A
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -bf-cbc -pass pass:shikhir -p -in plaintext.txt -out fileBlowFish.enc
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=2002BC9D0FD485E1
key=50FF5F28121D78E1A466C4452975025
iv =0672CEC29A59EBD24
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -aes-128-cfb -pass pass:shikhir -p -in plaintext.txt -out fileaes128cfb.enc
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=2E771AA4165DME14
key=FE3DD0F5A46062F7863F495FB173D0D
iv =72B2C017BA768A18A9389C6C12E635
shikher@shikhsvm:/home/seed/security/Labsetup$
```

```
shikher@shikhsvm: /home/seed/security/Labsetup - Google Chrome
ssh.cloud.google.com/projects/serene-cathode-329819/zones/us-central1-a/instances/shikhsvm?authuser=0&hl=en_US&projectNumber=588356155406&useAdminProxy=true&troubleshoot4005...
Verifying - enter aes-256-cbc encryption password:
Can't open dangle for writing, Permission denied
139798693045568:error:0200100D:system library:fopen:Permission denied:./crypto/b
io/bas file.c:69:fopen("dangle","w")
139798693045568:error:20060002:BIO routines:BIO_new_file:system lib:./crypto/b
io/bas file.c:78:
shikher@shikhsvm:/home/seed/security/Labsetup$ openssl list-cipher-commands
Invalid command 'list-cipher-commands'; type "help" for a list.
shikher@shikhsvm:/home/seed/security/Labsetup$ openssl enc -aes-256-cbc -pass
pass:shikhir -p -in plaintext.txt -out file.enc
Can't open file.enc for writing, Permission denied
140230302373568:error:0200100D:system library:fopen:Permission denied:./crypto/b
io/bas file.c:69:fopen("file.enc",'wb')
140230302373568:error:20060002:BIO routines:BIO_new_file:system lib:./crypto/b
io/bas file.c:78:
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -aes-256-cbc -
pass pass:shikhir -p -in plaintext.txt -out fileAES128.enc
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=F2F161C9A62
key=BSA10600E00X75A357D89A7F748393BCEDDBS108BD8E404E2041FFB8C6FB2B3
iv =1D9776FD024B68C766703AC0105A20
shikher@shikhsvm:/home/seed/security/Labsetup$ ls
Files      ciphertext.txt      encryption oracle      lowercase.txt      plaintext.txt
article.txt docker-compose.yml      file.enc      mysql_data
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -sha-256-cbc -pass pass:shikhir -p -in plaintext.txt -out fileSHA256.enc
enc: Unrecognized flag sha-256-cbc
enc: Use -help for summary.
shikher@shikhsvm:/home/seed/security/Labsetup$ -help
Command '-help' not found, did you mean:
  command 'dhelp' from deb dhelp (0.6.26)
Try: apt install <deb name>
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -aes-128-cbc -pass pass:shikhir -p -in plaintext.txt -out fileAES128.enc
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=F2F161C9A62
key=BSA10600E00X75A357D89A7F748393BCEDDBS108BD8E404E2041FFB8C6FB2B3
iv =1D9776FD024B68C766703AC0105A20
shikher@shikhsvm:/home/seed/security/Labsetup$ ls
Files      ciphertext.txt      encryption oracle      lowercase.txt      plaintext.txt
article.txt docker-compose.yml      file.enc      mysql_data
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -bf-cbc -pass pass:shikhir -p -in plaintext.txt -out fileBlowFish.enc
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=2002BC9D0FD485E1
key=50FF5F28121D78E1A466C4452975025
iv =0672CEC29A59EBD24
shikher@shikhsvm:/home/seed/security/Labsetup$ sudo openssl enc -aes-128-cfb -pass pass:shikhir -p -in plaintext.txt -out fileaes128cfb.enc
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=2E771AA4165DME14
key=FE3DD0F5A46062F7863F495FB173D0D
iv =72B2C017BA768A18A9389C6C12E635
shikher@shikhsvm:/home/seed/security/Labsetup$
```

The ciphers and modes used are AES-128-CFB, Blowfish-CBC, AES-256 and AES-128-CBC.

/home/seed/security/Labsetup/fileBlowfish.enc - Bless

fileBlowfish.enc

00000000	53 61 6C 74 65 64 5F 5F 20 02 BC 9D 0F D4 85 E1 A4	Salted_ .....
00000011	89 8D 75 DC C7 B4 58 78 03 06 02 5F 8D 09 95 FC 6C	.u...Xx..-....l
00000022	26 D1 DF 9D A7 5D 62 F9 46 DA A6 0D 81 9A 9B 6B 97	&....]b.F.....k.
00000033	C2 4B 3E 35 4B	K>5K

Signed 8 bit: 83      Signed 32 bit: 1398893684      Hexadecimal: 53 61 6C 74

Unsigned 8 bit: 83      Unsigned 32 bit: 1398893684      Decimal: 083 097 108 116

Signed 16 bit: 21345      Float 32 bit: 9.681872E+11      Octal: 123 141 154 164

Unsigned 16 bit: 21345      Float 64 bit: 4.54305331895921E+93      Binary: 01010011 01100001 01

Show little endian decoding     Show unsigned as hexadecimal    ASCII Text: Salt

Loaded file '/home/seed/sec...'    Offset: 0x0 / 0x37    Selection: None    INS

/home/seed/security/Labsetup/fileAES256.enc - Bless

fileAES256.enc

00000000	53 61 6C 74 65 64 5F 5F A0 BC 8F 6E C5 79 69 01 DA	Salted_...n.yi..
00000011	88 C5 1B 4D F8 4C 88 2F E1 0E 11 10 23 52 A1 B9 03	...M.L./....#R...
00000022	A8 C8 94 55 67 95 B0 8D 5F D9 E6 11 D5 95 9B 25 65	...Ug._....%e
00000033	7B DB 53 D3 D3 31 5F 74 D3 71 A2 AE E5	(.S..1_t.q...

Signed 8 bit: 83      Signed 32 bit: 1398893684      Hexadecimal: 53 61 6C 74

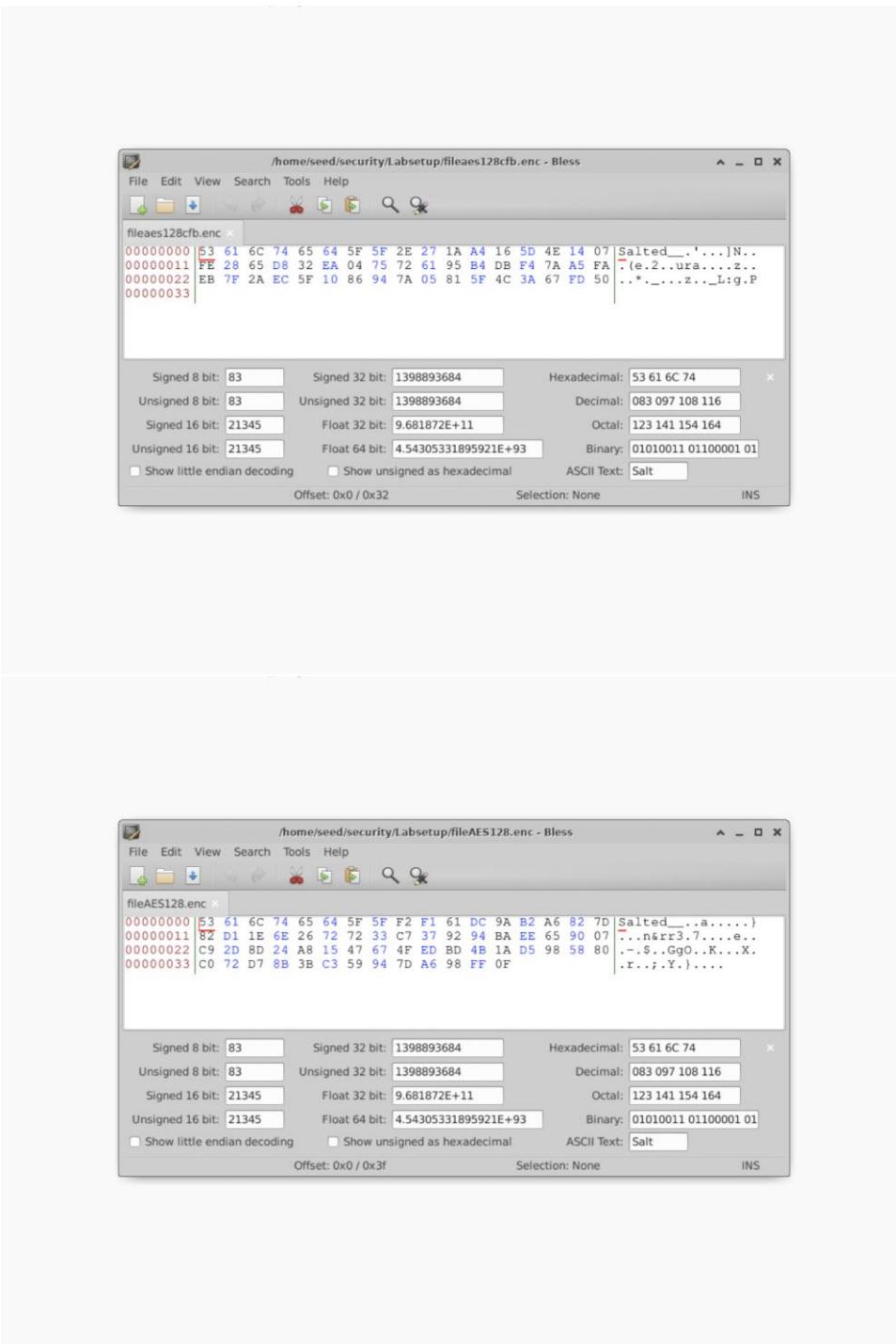
Unsigned 8 bit: 83      Unsigned 32 bit: 1398893684      Decimal: 083 097 108 116

Signed 16 bit: 21345      Float 32 bit: 9.681872E+11      Octal: 123 141 154 164

Unsigned 16 bit: 21345      Float 64 bit: 4.54305331895921E+93      Binary: 01010011 01100001 01

Show little endian decoding     Show unsigned as hexadecimal    ASCII Text: Salt

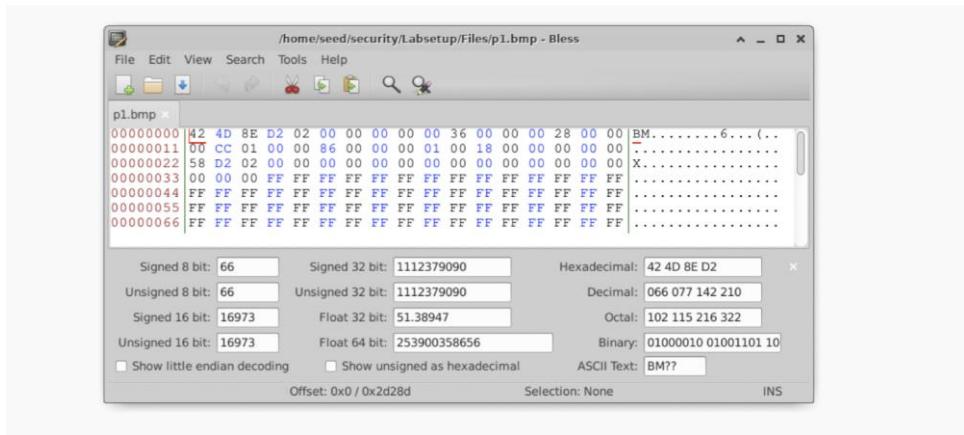
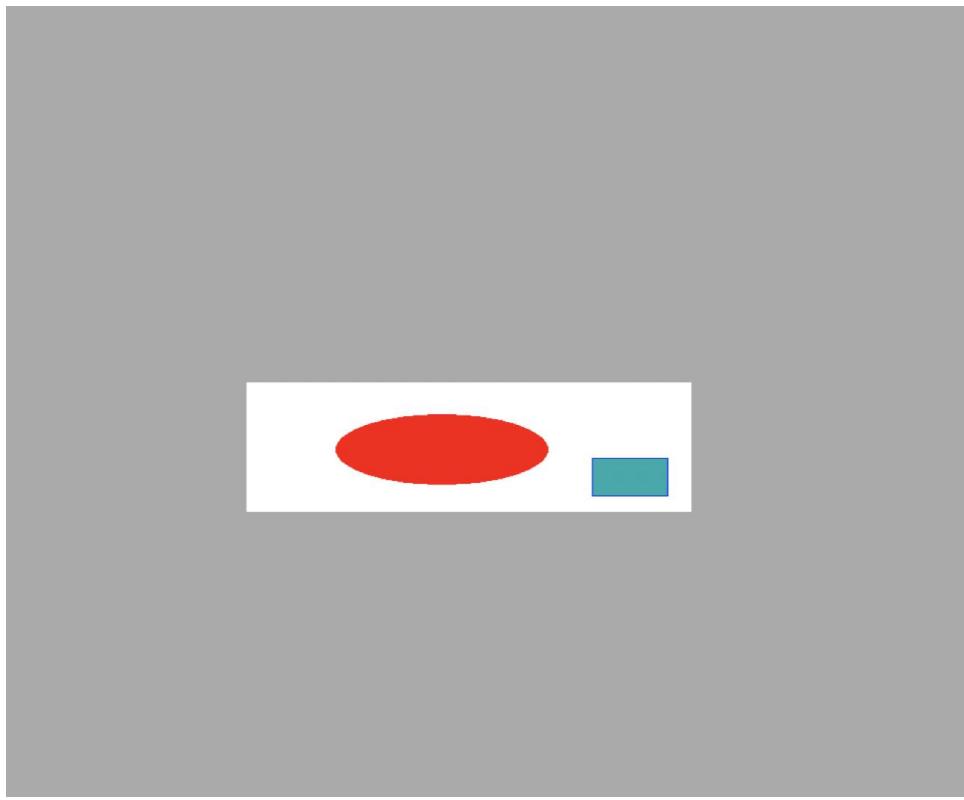
Offset: 0x0 / 0x3f    Selection: None    INS



I have also displayed the hexcode for the generated enc files. The sizes of the files may vary depending on the cipher and the mode.

We observe that we have successfully encrypted the files using different ciphers and modes.

### Task 3: Encryption Mode – ECB vs. CBC

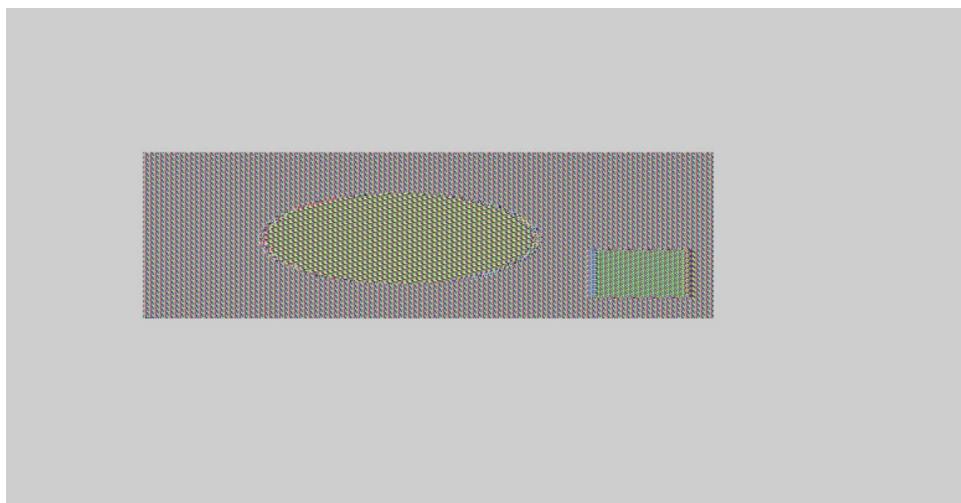


We first look at the original picture given to us; I have renamed it to p1 for the ease of use. We are going to encrypt it using different modes and use the original 54 bytes from the unencrypted image as the header for the bmp file and the body from the encrypted file.

## **1) Encrypting the image using ECB (Electronic Code Book)**

Using the openssl command we encrypt the image with aes-128-ecb and use the 55+ bytes of the file as the body of the encrypted image. The header comes from the original image file.

```
seed@shikhirsrvm: ~/security/Labsetup/Files
File Edit View Search Terminal Help
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -e -in p1.bmp
p -out plecb.bmp
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ head -c 54 p1.bmp > header
seed@shikhirsrvm:~/security/Labsetup/Files$ tail -c +55 plecb.bmp > body
seed@shikhirsrvm:~/security/Labsetup/Files$ cat header body > piclecb.bmp
seed@shikhirsrvm:~/security/Labsetup/Files$ ls piclecb.bmp
piclecb.bmp
seed@shikhirsrvm:~/security/Labsetup/Files$
```



Here we can see that we have successfully redirected the output to the bmp file. The first 54 bits of the encrypted bmp file are replaced with the first 54 bits of the original bmp file. The encrypted image can be seen using image viewer. Information about the original image can be observed in this encrypted bmp file because ECB mode generates same cipher text for repeating plain text. Even though the cipher image is different from the original image, most of the original image's information can be obtained from the cipher image and the shapes are preserved.

## 2) Encrypting the image using CBC (Cipher block chaining)

Block cipher encryption - each block of the plain text is XORed with the previously generated cipher text and then it is encrypted. So, each cipher block depends upon all plain texts that were processed.

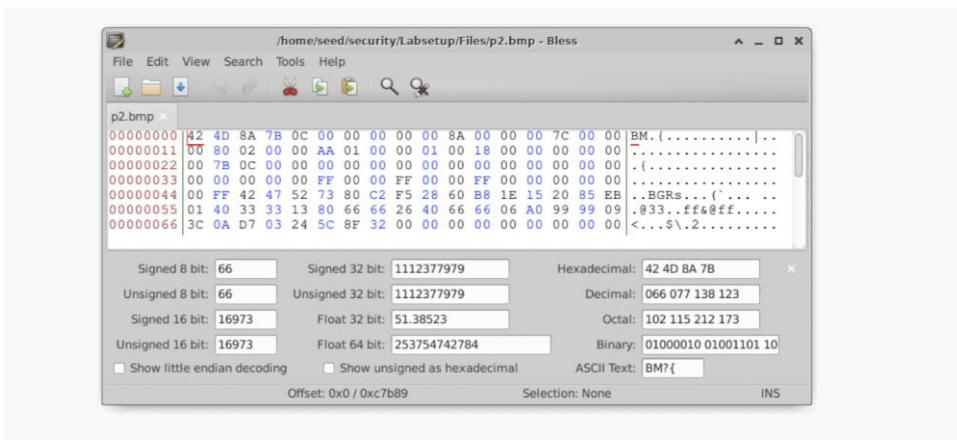
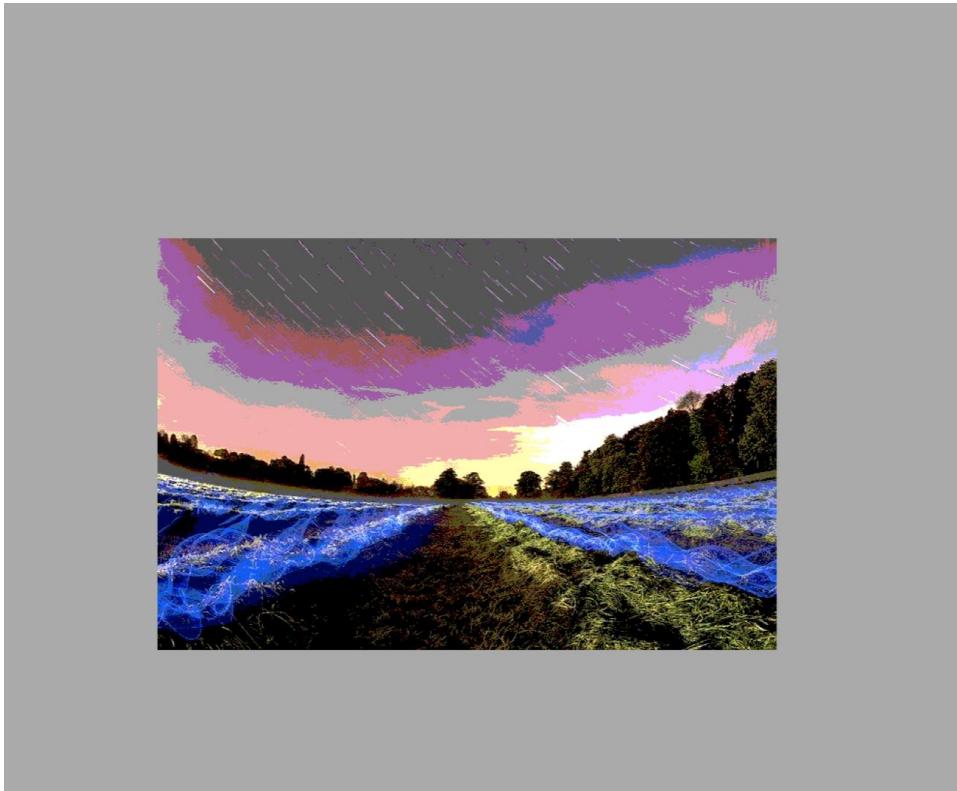
```
seed@shikhirsrvm:~/security/Labsetup/Files$ cat header body > pic1cbc.bmp
seed@shikhirsrvm:~/security/Labsetup/Files$ ls pic1cbc.bmp
pic1cbc.bmp
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -e -in pl.bmp
-p-out plcbc.bmp
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -e -in pl.bmp
-p-out plcbc.bmp
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ head -c 54 pl.bmp > header
seed@shikhirsrvm:~/security/Labsetup/Files$ tail -c +55 plcbc.bmp > body
seed@shikhirsrvm:~/security/Labsetup/Files$ cat header body > pic1cbc.bmp
seed@shikhirsrvm:~/security/Labsetup/Files$ ls pic1cbc.bmp
pic1cbc.bmp
seed@shikhirsrvm:~/security/Labsetup/Files$
```

We use the openssl AES-CBC mode to encrypt the original bmp file.



The first 54 bits of the encrypted bmp file are replaced with the first 54 bits of the original bmp file. However, any information about the original image cannot be observed in this encrypted bmp file because CBC mode generates different cipher text for repeating plain text.

**Trying the same experiment use our own sample bmp image**



## 1) Encrypting the image using ECB (Electronic Code Book)

Using the openssl command we encrypt the image with aes-128-ecb and use the 55+ bytes of the file as the body of the encrypted image. The header comes from the original image file.

```
File Edit View Search Tools Help
seed@shikhirsrvm: ~/security/Labsetup/Files
File Edit View Search Terminal Help
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ head -c 54 p1.bmp > header
seed@shikhirsrvm:~/security/Labsetup/Files$ tail -c +55 plcbc.bmp > body
seed@shikhirsrvm:~/security/Labsetup/Files$ cat header body > pic1cbc.bmp
seed@shikhirsrvm:~/security/Labsetup/Files$ ls pic1cbc.bmp
pic1cbc.bmp
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -e -in p2.bmp
p -out p2ecb.bmp
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -e -in p2.bmp
p -out p2ecb.bmp
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ head -c 54 p2.bmp > header
seed@shikhirsrvm:~/security/Labsetup/Files$ tail -c +55 p2ecb.bmp > body
seed@shikhirsrvm:~/security/Labsetup/Files$ cat header body > pic2ecb.bmp
seed@shikhirsrvm:~/security/Labsetup/Files$
```

We use the openssl AES-ECB mode to encrypt the original bmp file.

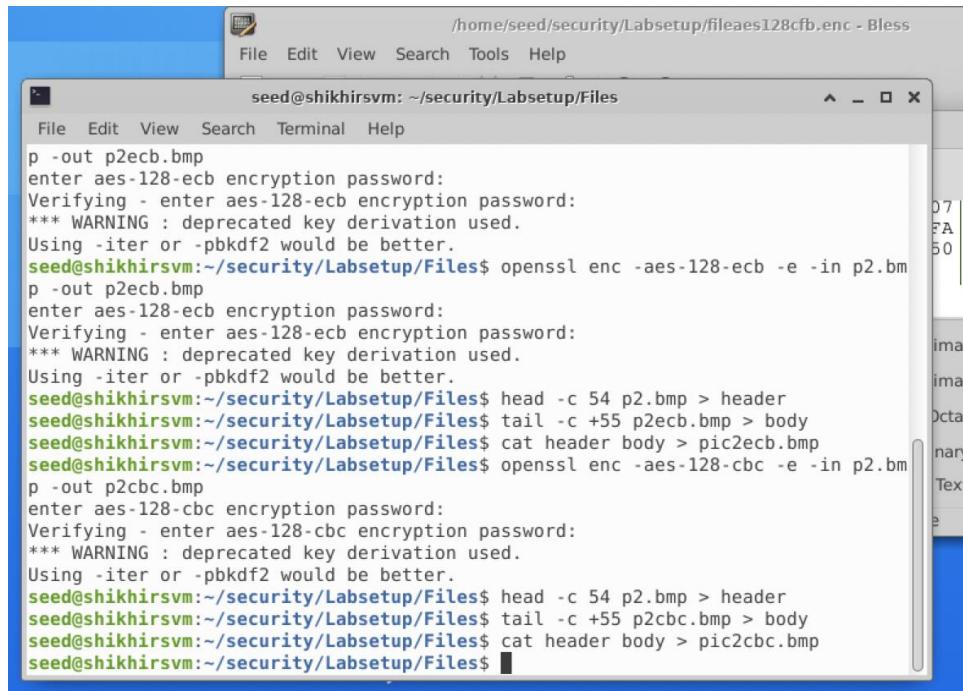


Here we can see that we have successfully redirected the output to the bmp file. The first 54 bits of the encrypted bmp file are replaced with the first 54 bits of the original bmp file. The encrypted image can be seen using image viewer. Information about the original image can be observed in this encrypted bmp file because ECB mode generates same cipher text for repeating plain text. Even though the cipher image is different from the original image,

most of the original image's information can be obtained from the cipher image and the shapes are preserved.

## 2) Encrypting the image using CBC (Cipher block chaining)

Block cipher encryption - each block of the plain text is XORed with the previously generated cipher text and then it is encrypted. So, each cipher block depends upon all plain texts that were processed.



The screenshot shows a terminal window titled 'seed@shikhirsvm: ~/security/Labsetup/Files'. The terminal displays the following command sequence:

```
p -out p2ecb.bmp
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -e -in p2.bmp
p -out p2ecb.bmp
enter aes-128-ecb encryption password:
Verifying - enter aes-128-ecb encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsvm:~/security/Labsetup/Files$ head -c 54 p2.bmp > header
seed@shikhirsvm:~/security/Labsetup/Files$ tail -c +55 p2ecb.bmp > body
seed@shikhirsvm:~/security/Labsetup/Files$ cat header body > pic2ecb.bmp
seed@shikhirsvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -e -in p2.bmp
p -out p2cbc.bmp
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsvm:~/security/Labsetup/Files$ head -c 54 p2.bmp > header
seed@shikhirsvm:~/security/Labsetup/Files$ tail -c +55 p2cbc.bmp > body
seed@shikhirsvm:~/security/Labsetup/Files$ cat header body > pic2cbc.bmp
seed@shikhirsvm:~/security/Labsetup/Files$
```

We use the openssl AES-CBC mode to encrypt the original bmp file.



The first 54 bits of the encrypted bmp file are replaced with the first 54 bits of the original bmp file. However, any information about the original image cannot be observed in this encrypted bmp file because CBC mode generates different cipher text for repeating plain text. Hence, we can conclude that the original shape in the image is not preserved.

## **Task 4: Padding**

We check for the Encryptions that use padding mechanisms to encrypt.

We first create 3 files of the size 5, 10, 16 bytes respectively and then encrypt them using different modes, we then decrypt them to check if there is some padding added to the file.

### **1) Encryption and Decryption using CBC**

We encrypt all the 3 files using the CBC method, the size of the files is 32 bytes.

```
seed@shikhirsrvm: ~/security/Labsetup/Files$ 
File Edit View Search Terminal Help
seed@shikhirsrvm:~/security/Labsetup/Files$ echo -n "12345" > f5.txt
seed@shikhirsrvm:~/security/Labsetup/Files$ echo -n "12345">f5.txt
seed@shikhirsrvm:~/security/Labsetup/Files$ echo -n "1234567890">f10.txt
seed@shikhirsrvm:~/security/Labsetup/Files$ echo -n "1234567890123456">f16.txt
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -e -in f5.txt -out fe5.enc
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -e -in f10.txt -out fe10.enc
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -e -in f16.txt -out fe16.enc
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$
```

27 Bytes are appended to the 5 Byte file as padding.

22 Bytes are appended to the 10 Byte file as padding.

16 Bytes are appended to the 16 Byte file as padding.

```
seed@shikhirsrvm:~/security/Labsetup/Files$ 
File Edit View Search Terminal Help
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -e -in f10.txt -out fe10.enc
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -e -in f16.txt -out fe16.enc
enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5.txt
00000000: 3132 3334 12345
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe5.enc -out f05
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f05.txt
xxd: f05.txt: No such file or directory
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe5.enc -out f05.txt
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f05.txt
00000000: 3132 3334 12345
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe10.enc -out f010.txt
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f010.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe5.enc -out f05.txt -nopad
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5.txt
00000000: 3132 3334 12345
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f05.txt
00000000: 3132 3334 350b 0b0b 0b0b 0b0b 12345..... .
seed@shikhirsrvm:~/security/Labsetup/Files$
```

```
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f05.txt
xxd: f05.txt: No such file or directory
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe5.enc -out f05.txt
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f05.txt
00000000: 3132 3334 3536 3738 3930 12345
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe10.enc -out f010.txt
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f010.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe5.enc -out f05.txt -nopad
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5.txt
00000000: 3132 3334 3536 3738 3930 12345
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f05.txt
00000000: 3132 3334 3536 3738 3930 1234567890.....
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe10.enc -out f010.txt -nopad
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890.....
seed@shikhirsrvm:~/security/Labsetup/Files$
```

```
seed@shikhirsrvm:~/security/Labsetup/Files$ File Edit View Search Terminal Help
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f010.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe5.enc -out f05.txt -nopad
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5.txt
00000000: 3132 3334 3536 3738 3930 12345
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f05.txt
00000000: 3132 3334 3536 3738 3930 1234567890.....
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe10.enc -out f010.txt -nopad
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f010.txt
00000000: 3132 3334 3536 3738 3930 1234567890.....
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe10.enc -out f11.txt -nopad
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f16.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f16.txt
00000000: 3132 3334 3536 3738 3930 1234567890.....
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cbc -d -in fe16.enc -out f016.txt -nopad
enter aes-128-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f16.txt
00000000: 3132 3334 3536 3738 3930 3132 3334 3536 1234567890123456
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f016.txt
00000000: 3132 3334 3536 3738 3930 3132 3334 3536 1234567890123456
00000010: 1010 1010 1010 1010 1010 1010 1010 1010 .....
seed@shikhirsrvm:~/security/Labsetup/Files$
```

The plain text is encrypted using AES-128-bit cipher with CBC mode. Padding is performed before the encryption starts.

## 2) Encryption and Decryption using CFB

The plain text for all the 3 files is encrypted using AES-128-bit cipher with CFB mode.

```

eed@shikhirsrvm:~/security/Labsetup/Files$ 
eed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cfb -e -in f5.txt -out f5cbc.bin -pass pass:123
** WARNING : deprecated key derivation used.
sing -iter or -pbkdf2 would be better.
eed@shikhirsrvm:~/security/Labsetup/Files$ ls f5cbc.bin -lh
rw-rw-r-- 1 seed seed 21 Nov 12 10:37 f5cbc.bin
eed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cfb -d -in f5cbc.bin -out f5cfb.txt -pass pass:123
** WARNING : deprecated key derivation used.
sing -iter or -pbkdf2 would be better.
eed@shikhirsrvm:~/security/Labsetup/Files$ ls f5cfb.txt -lh
rw-rw-r-- 1 seed seed 5 Nov 12 10:40 f5cfb.txt
eed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5cbc.bin
00000000: 5361 6c74 6564 5f5f 3f64 4339 bcd4 6a7b Salted__?dC9..j{
00000010: ee39 e6ec a3 .9...
eed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5cfb.txt
00000000: 3132 3334 35 12345
eed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5cfb.bin
xd: f5cfb.bin: No such file or directory
eed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5cfb.txt
00000000: 3132 3334 35 12345
eed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cfb -d -in f5cbc.bin -out f5cfb.txt -pass pass:123 -nopad
** WARNING : deprecated key derivation used.
sing -iter or -pbkdf2 would be better.
eed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5cfb.txt
00000000: 3132 3334 35 12345
eed@shikhirsrvm:~/security/Labsetup/Files$ 

```

```

00000000: 3132 3334 35 12345
eed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cfb -e -in f10.txt -out f10cfb.bin -pass pass:123
** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ ls f10cfb.bin -lh
rw-rw-r-- 1 seed seed 26 Nov 12 10:44 f10cfb.bin
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cfb -d -in f10cfb.bin -out f10cfb.txt -pass pass:123 -nopad
** WARNING : deprecated key derivation used.
sing -iter or -pbkdf2 would be better.
eed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10cfb.txt
00000000: 3132 3334 35 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10cfb.bin
00000000: 5361 6c74 6564 5f5f 8544 9edf 2344 4425 Salted__D..#00%
00000010: 65a4 f4da 202b 3070 c4d e...+0p.M
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cfb -d -in f10cfb.bin -out f10cfb.txt -pass pass:123 -nopad
** WARNING : deprecated key derivation used.
sing -iter or -pbkdf2 would be better.
eed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10cfb.txt
00000000: 3132 3334 35 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ 

```

```

00000000: 5361 6c74 6564 5f5f 8544 9edf 2344 4425 Salted__D..#00%
00000010: 65a4 f4da 202b 3070 c4d e...+0p.M
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cfb -d -in f10cfb.bin -out f10cfb.txt -pass pass:123 -nopad
** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10cfb.txt
00000000: 5361 6c74 6564 5f5f 8544 9edf 2344 4425 Salted__D..#00%
00000010: 65a4 f4da 202b 3070 c4d e...+0p.M
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-cfb -d -in f10cfb.bin -out f10dcfb.txt -pass pass:123 -nopad
** WARNING : deprecated key derivation used.
sing -iter or -pbkdf2 would be better.
eed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10dcfb.txt
00000000: 3132 3334 35 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ 

```

We encrypt the files using the CFB mode of encryption. The encrypted file size varies according to the size of the original text file (i.e., 21, 26, 32 bytes). When we decrypt (with “-nopad” argument), we get the exact same text as our original plaintext. Hence, no padding is done.

### 3) Encryption and Decryption using OFB

Perform encryption using OFB mode. We get varying sizes of the encrypted file.

```

00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ofb -e -in f16.txt -out f16ofb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ ls f16ofb.bin -lh
-rw-r--r-- 1 seed seed 32 Nov 12 10:56 f16ofb.bin
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ofb -d -in f16ofb.bin -out f16dofb.txt -pass pass:123 -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f16.txt
00000000: 3132 3334 3536 3738 3930 3132 3334 3536 1234567890123456
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f16dofb.txt
00000000: 3132 3334 3536 3738 3930 3132 3334 3536 1234567890123456
seed@shikhirsrvm:~/security/Labsetup/Files$ 

```

```

seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ofb -e -in f10.txt -out f10ofb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ ls f10ofb.bin -lh
-rw-r--r-- 1 seed seed 26 Nov 12 10:54 f10ofb.bin
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ofb -d -in f10ofb.bin -out f10dofb.txt -pass pass:123 -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10dofb.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ 

```

```

seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f16dcfb.txt
00000000: 3132 3334 3536 3738 3930 3132 3334 3536 1234567890123456
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ofb -e -in f5.txt -out f5ofb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ ls f5ofb.bin -lh
-rw-r--r-- 1 seed seed 21 Nov 12 10:51 f5ofb.bin
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ofb -d -in f5ofb.bin -out f5dofb.txt -pass pass:123 -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5.txt
00000000: 3132 3334 3536 3738 3930 12345
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5dofb.txt
00000000: 3132 3334 3536 3738 3930 12345
seed@shikhirsrvm:~/security/Labsetup/Files$ 

```

When we decrypt with “-nopad” argument, we get the same text as our original plain text. Hence, no padding is done.

## 4) Encryption and Decryption using ECB

In ECB, we get greater size of encrypted files than OFB and CFB modes.

```

00000000: 3132 3334 3536 0000 0000 0000 0000 12345..... .
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -e -in f10.txt -out f10ecb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ ls f10ecb.bin -lh
-rw-r--r-- 1 seed seed 32 Nov 12 11:03 f10ecb.bin
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -d -in f10ecb.bin -out f510ecb.txt -pass pass:123 -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd 105decb.txt
xxd: 105decb.txt: No such file or directory
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd 10decb.txt
xxd: 10decb.txt: No such file or directory
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f510decb.txt
xxd: f510decb.txt: No such file or directory
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f510ecb.txt
00000000: 3132 3334 3536 3738 3930 0660 0606 0606 1234567890.....
seed@shikhirsrvm:~/security/Labsetup/Files$ 

```

```

seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -e -in f5.txt -out f5ecbb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -e -in f5.txt -out f5ecb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ ls f5ecb.bin -lh
-rw-r--r-- 1 seed seed 30 Nov 12 11:00 f5ecb.bin
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ofb -d -in f5ecb.bin -out f5decb.txt -pass pass:123 -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5.txt
00000000: 3132 3334 3536 3738 3930 12345
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5decb.txt
xxd: f5ecb.txt: No such file or directory
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5decb.txt
00000000: 6081 file 6d42 722e efe5 50bf 3ecb 21de `..MBR...P,>.!
seed@shikhirsrvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -d -in f5ecb.bin -out f5decb.txt -pass pass:123 -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup/Files$ xxd f5decb.txt
00000000: 3132 3334 3536 0000 0000 0000 0000 12345..... .
seed@shikhirsrvm:~/security/Labsetup/Files$ 

```

```

xxd -r /root/Desktop/seed/shikhirsvm:/security/Labsetup/Files$ xxd f510decb.txt
xxd: f510decb.txt: No such file or directory
seed@shikhirsvm:~/security/Labsetup/Files$ xxd f10.txt
00000000: 3132 3334 3536 3738 3930 1234567890
seed@shikhirsvm:~/security/Labsetup/Files$ xxd f510ecb.txt
00000000: 3132 3334 3536 3738 3930 0606 0606 1234567890.....
seed@shikhirsvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -e -in f10.txt -out f16ecb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsvm:~/security/Labsetup/Files$ ls f16ecb.bin -lh
-rw-r--r-- 1 seed seed 48 Nov 12 11:07 f16ecb.bin
seed@shikhirsvm:~/security/Labsetup/Files$ 
seed@shikhirsvm:~/security/Labsetup/Files$ openssl enc -aes-128-ecb -d -in f16ecb.bin -out f16ecb.txt -pass pass:123 -nopad
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsvm:~/security/Labsetup/Files$ xxd f16.txt
00000000: 3132 3334 3536 3738 3930 3132 3334 3536 1234567890123456
seed@shikhirsvm:~/security/Labsetup/Files$ xxd f16ecb.txt
00000000: 3132 3334 3536 3738 3930 3132 3334 3536 1234567890123456
00000010: 1010 1010 1010 1010 1010 1010 1010 1010 .....
seed@shikhirsvm:~/security/Labsetup/Files$ 

```

On Decrypting we see a “.” character which suggests that padding is used in ECB encryption mode.

In CFB and OFB, padding is not required as plaintext is XORed with the output of previous cipher block. The first x-bytes of key stream of previous block is XORed with the last x bytes of plaintext, producing x-bytes of cipher text.

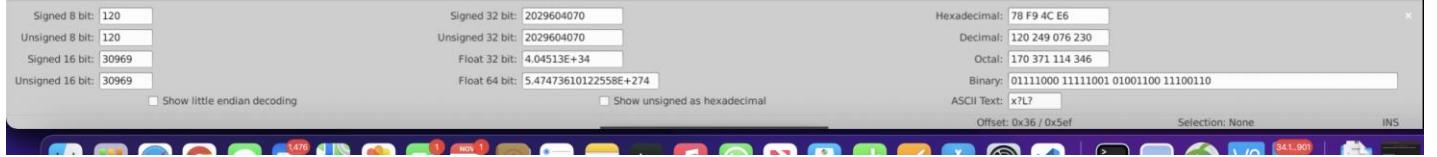
## **Task 5: Error Propagation – Corrupted Cipher Text**

Generate cipher from a random paragraph, change the 55<sup>th</sup> byte and notice its effect on the decrypted text.

```
seed@shikhirsrvm: ~/security/Labsetup
File Edit View Search Terminal Help
seed@shikhirsrvm:~/security/Labsetup$ cat largefile.txt
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

## Encryption and decryption using cbc

Signed 8 bit:	<input type="text" value="83"/>	Signed 32 bit:	<input type="text" value="1398893684"/>	Hexadecimal:	<input type="text" value="53 61 EC 74"/>
Unsigned 8 bit:	<input type="text" value="83"/>	Unsigned 32 bit:	<input type="text" value="1398893684"/>	Decimal:	<input type="text" value="083 097 108 116"/>
Signed 16 bit:	<input type="text" value="21345"/>	Float 32 bit:	<input type="text" value="9.681872E+11"/>	Octal:	<input type="text" value="123 141 154 164"/>
Unsigned 16 bit:	<input type="text" value="21345"/>	Float 64 bit:	<input type="text" value="4.5430533189592E+93"/>	Binary:	<input type="text" value="01010011 01100001 01101100 01110100"/>
<input type="checkbox"/> Show little endian decoding		<input type="checkbox"/> Show unsigned as hexadecimal		ASCII Text:	<input type="text" value="Salt"/>
			Offset:	<input type="text" value="0x0 / 0x5ef"/>	Selection: None
INS					



Modifying 55<sup>th</sup> byte and decrypting.

## **Output for decrypted cbc file**

By changing the 55<sup>th</sup> byte, the decryption has significant changes. It completely distorts the text in a particular region and the transmitted file may not be readable.

## Encrypting the same file using different modes – cfb, ecb and ofb

```
seed@shikhirsrvm: ~/security/Labsetup
File Edit View Search Terminal Help
fe5.enc new.bmp ptext.txt sample_code.py
seed@shikhirsrvm:~/security/Labsetup$ ls
body f10cfb.bin f16.txt f5.txt f5ecbb.bin new.bmp words.txt
ciphertext.txt f10cfb.txt f16fb.bin f510ecb.txt f5ofb.bin p1.bmp
f010.txt f10dcfb.txt f16dcfb.txt f5cbc.bin fe10.enc p2.bmp
f016.txt f10dofb.txt f16dofb.txt f5cfb.txt fe16.enc ptext.txt
f05 f10ecb.bin f16ecb.bin f5decg.txt fe5.enc s1.bmp
f05.txt f10ofb.bin f16ecb.txt f5dofb.txt freq.py s2.bmp
f10.txt f11.txt f16ofb.bin f5ecb.bin header sample_code.py
seed@shikhirsrvm:~/security/Labsetup$ cd ..
seed@shikhirsrvm:~/security/Labsetup$ ls
Files docker-compose.yml file.enc fileBlowfish.enc lowercase.txt
article.txt encr.bin fileAES128.enc fileaes128cfb.enc mysql_data
ciphertext.txt encryption_oracle fileAES256.enc largefile.txt plaintext.txt
seed@shikhirsrvm:~/security/Labsetup$ openssl enc -aes-128-ecb -e -in largefile.txt -out lfecb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup$ openssl enc -aes-128-ofc -e -in largefile.txt -out lfocf.cbin -pass pass:123
enc: Unrecognized flag aes-128-ofc
enc: Use -help for summary.
seed@shikhirsrvm:~/security/Labsetup$ openssl enc -aes-128-ofb -e -in largefile.txt -out lfofb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup$ openssl enc -aes-128-cfb -e -in largefile.txt -out lfcb.bin -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsrvm:~/security/Labsetup$
```

## Corrupting the hexcode of the cfb bin file – using bless

Signed 8 bit: -100	Signed 32 bit: -1670202389	Hexadecimal: 9C 72 B8 EB
Unsigned 8 bit: 156	Unsigned 32 bit: 2624764907	Decimal: 156 114 187 235
Signed 16 bit: -25486	Float 32 bit: -8.031396E-22	Octal: 234 162 273 353

## Corrupting the hexcode of the ecb bin file – using bless

Signed 8 bit: -30	Signed 32 bit: -501112776	Hexadecimal: E2 21 A0 38
Unsigned 8 bit: 226	Unsigned 32 bit: 3793854520	Decimal: 226 033 160 056
Signed 16 bit: -7647	Float 32 bit: -7.453677E+20	Octal: 342 041 240 070

Signed 8 bit:	50	Signed 32 bit:	841064504	Hexadecimal:	32 21 A0 38
Unsigned 8 bit:	50	Unsigned 32 bit:	841064504	Decimal:	050 033 160 056
Signed 16 bit:	12833	Float 32 bit:	9.407863E-09	Octal:	062 041 240 070

## Corrupting the hexcode of the ofb bin file – using bless

Signed 8 bit:	-16	Signed 32 bit:	-254325214	Hexadecimal:	F0 D7 4E 22
Unsigned 8 bit:	240	Unsigned 32 bit:	4040642082	Decimal:	240 215 078 034
Signed 16 bit:	-3881	Float 32 bit:	-5.330699E+29	Octal:	360 327 116 042

## Decrypting all files using different modes

```
seed@shikhirsvm:~/security/Labsetup$ openssl enc -aes-128-ecb -d -in lfecb.bin -out lfecb1.txt -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsvm:~/security/Labsetup$ openssl enc -aes-128-ofb -d -in lfofb.bin -out lfofb1.txt -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsvm:~/security/Labsetup$ openssl enc -aes-128-cfb -d -in lfcfb.bin -out lfcfb1.txt -pass pass:123
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
seed@shikhirsvm:~/security/Labsetup$
```

## **Output for decrypted cfb file**

By changing the 55<sup>th</sup> byte, the decryption has significant changes. It completely distorts the text in a particular region and the transmitted file may not be readable.

## **Output for decrypted ofb file**

By changing the 55<sup>th</sup> byte, the decryption has very minor changes. It distorts the text byte in a particular region.

## **Output for decrypted ecb file**

By changing one byte, the decryption has very noticeable changes. It completely distorts the text in a particular region.

