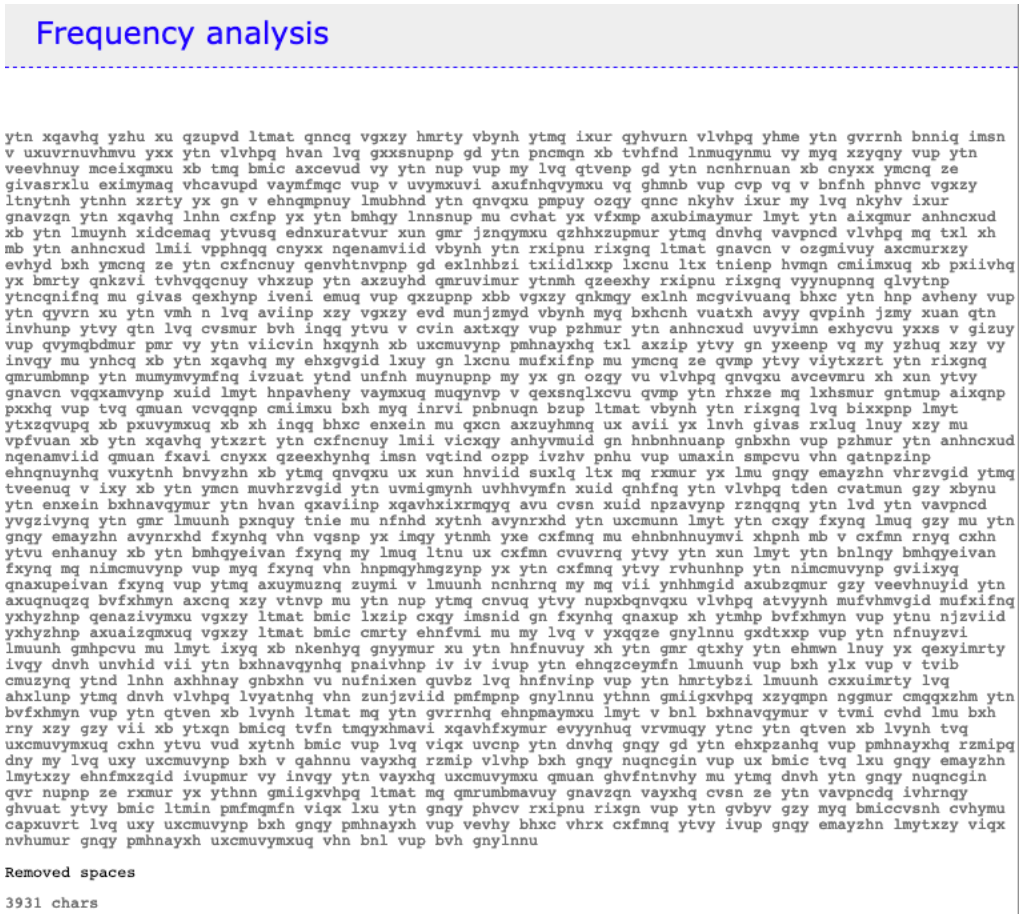
Name: Bharath Karumudi
Lab: Secret-Key Encryption

Task 1: Frequency Analysis Against Monoalphabetic Substitution Cipher



Frequency analysis

ytn xqavhq yzhu xu qzupvd ltmat qnncq vgxzy hmrty vbynh ytmq ixur qyhvurn vlvhpq yhme ytn gvrrnh bnniq imsn
v uxuvrnuvhmvu yxx ytn vlvhpq hvan lvq gxxsnupnp gd ytn pncmqn xb tvhfnd lnmuqynmu vy myq xzyqny vup ytn
veevhnuy mceixqmxu xb tmq bmic axcevud vy ytn nup vup my lvq qtvenp gd ytn ncnhrnuan xb cnyxx ymcnq ze
givasrxlu eximymaq vhcavupd vaymfmqc vup v uvymxuvi axufnhqvymxu vq ghmnb vup cvp vq v bnfnh phnvc vgxzy
ltnytnh ytnhn xzrty yx gn v ehnqmpnuy lmubhnd ytn qnvqxu pmpuy ozqy qnnc nkyhv ixur my lvq nkyhv ixur
gnavzqn ytn xqavhq lnhn cxfnp yx ytn bmhqy lnnsnup mu cvhat yx vfxmp axubimaymur lmyt ytn aixqmur anhncxud
xb ytn lmuynh xidcemaq ytvusq ednxuratvur xun gmr jznqymxu qzhhxzupmur ytmq dnvhq vavpncd vlvhpq mq txl xh
mb ytn anhncxud lmii vpphnqq cnyxx nqenamviid vbynh ytn rxipnu rixgnq ltmat gnavcn v ozgmivuy axcmurxzy
evhyd bxh ymcnq ze ytn cxfncnuy qenvhtnvpnp gd exlnhbzi txiidlxxp lxcnu ltx tnienp hvmqn cmiimxuq xb pxiivhq
yx bmrty qnkzvi tvhvqqcnuy vhxzup ytn axzuyhd qmruvimur ytnmh qzeexhy rxipnu rixgnq vyynupnnq qlvytnp
ytncqnifnq mu givas qexhynp iveni emuq vup qxzupnp xbb vgxzy qnkmqy exlnh mcgvivuanq bhxc ytn hnp avheny vup
ytn qyvrn xu ytn vmh n lvq aviinp xzy vgxzy evd munjzmyd vbynh myq bxhcnh vuatxh avyy qvpinh jzmy xuan qtn
invhunp ytvy qtn lvq cvsmur bvh inqq ytvu v cvin axtxqy vup pzhmur ytn anhncxud uvyvimn exhycvu yxxs v gizuy
vup qvymqbdmur pmr vy ytn viicvin hxqynh xb uxcmuvynp pmhnayxhq txl axzip ytvy gn yxeenp vq my yzhuq xzy vy
invqy mu ynhcq xb ytn xqavhq my ehxgvgid lxuy gn lxcnu mufxifnp mu ymcnq ze qvmp ytvy viytxzrt ytn rixgnq
qmrumbmnp ytn mumymvymfnq ivzuat ytnd unfnh muynupnp my yx gn ozqy vu vlvhpq qnvqxu avcevmru xh xun ytvy
gnavcn vqqxamvynp xuid lmyt hnpavheny vaymxuq muqynvp v qexsnqlxcvu qvmp ytn rhxze mq lxhsmur gntmup aixqnp
pxxhq vup tvq qmuan vcvqqnp cmiimxu bxh myq inrvi pnbnuqn bzup ltmat vbynh ytn rixgnq lvq bixxpnp lmyt
ytxzqvupq xb pxuvymxuq xb xh inqq bhxc enxein mu qxcn axzuyhmnq ux avii yx lnvh givas rxluq lnuy xzy mu
vpfvuan xb ytn xqavhq ytxzrt ytn cxfncnuy lmii vicxqy anhyvmuid gn hnbnhnuanp gnbxhn vup pzhmur ytn anhncxud
nqenamviid qmuan fxavi cnyxx qzeexhynhq imsn vqtind ozpp ivzhv pnhu vup umaxin smpcvu vhn qatnpzinp
ehnqnuynhq vuxytnh bnvyzhn xb ytmq qnvqxu ux xun hnviid suxlq ltx mq rxmur yx lmu gnqy emayzhn vhrzvgid ytmq
tveenuq v ixy xb ytn ymcn muvhrzvgid ytn uvmigmynh uvhhvymfn xuid qnhfnq ytn vlvhpq tden cvatmun gzy xbynu
ytn enxein bxhnavqymur ytn hvan qxaviinp xqavhxixrmqyq avu cvsn xuid npzavynp rznqqnq ytn lvd ytn vavpncd
yvgzivynq ytn gmr lmuunh pxnquy tnie mu nfnhd xytnh avynrxhd ytn uxcmunn lmyt ytn cxqy fxynq lmuq gzy mu ytn
gnqy emayzhn avynrxhd fxynhq vhn vqsnp yx imqy ytnmh yxe cxfmnq mu ehnbnhnuymvi xhpnh mb v cxfmn rnyq cxhn
ytvu enhanuy xb ytn bmhqyeivan fxynq my lmuq ltnu ux cxfmn cvuvrnq ytvy ytn xun lmyt ytn bnlnqy bmhqyeivan
fxynq mq nimcmuvynp vup myq fxynq vhn hnpmqyhmgzynp yx ytn cxfmnq ytvy rvhunhnp ytn nimcmuvynp gviixyq
qnaxupeivan fxynq vup ytmq axuymuznq zuymi v lmuunh ncnhrnq my mq vii ynhhmgid axubzqmur gzy veevhnuyid ytn
axuqnuqzq bvfxhmyn axcnq xzy vtnvp mu ytn nup ytmq cnvuq ytvy nupxbqnvqxu vlvhpq atvyynh mufvhmvgid mufxifnq
yxhyzhnp qenazivymxu vgxzy ltmat bmic lxzip cxqy imsnid gn fxynhq qnaxup xh ytmhp bvfxhmyn vup ytnu njzviid
yxhyzhnp axuaizqmxuq vgxzy ltmat bmic cmrty ehnfvmi mu my lvq v yxqqze gnylnnu gxdtxxp vup ytn nfnuyzvi
lmuunh gmhpcvu mu lmyt ixyq xb nkenhyq gnyymur xu ytn hnfnuvuy xh ytn gmr qtxhy ytn ehmwn lnuy yx qexyimrty
ivqy dnvh unvhid vii ytn bxhnavqynhq pnaivhnp iv iv ivup ytn ehnqzceymfn lmuunh vup bxh ylx vup v tvib
cmuzynq ytnd lnhn axhhnay gnbxhn vu nufnixen quvbz lvq hnfnvinp vup ytn hmrtybzi lmuunh cxxuimrty lvq
ahxlunp ytmq dnvh vlvhpq lvyatnhq vhn zunjzviid pmfmpnp gnylnnu ythnn gmiigxvhpq xzyqmpn nggmur cmqqxzhm ytn
bvfxhmyn vup ytn qtven xb lvynh ltmat mq ytn gvrrnhq ehnpmaymxu lmyt v bnl bxhnavqymur v tvmi cvhd lmu bxh
rny xzy gzy vii xb ytxqn bmicq tvfn tmqyxhmavi xqavhfxymur evyynhuq vrvmuqy ytnc ytn qtven xb lvynh tvq
uxcmuvymxuq cxhn ytvu vud xytnh bmic vup lvq viqx uvcnp ytn dnvhq gnqy gd ytn ehxpzanhq vup pmhnayxhq rzmipq
dny my lvq uxy uxcmuvynp bxh v qahnnu vayxhq rzmip vlvhp bxh gnqy nuqncgin vup ux bmic tvq lxu gnqy emayzhn
lmytxzy ehnfmxzqid ivupmur vy invqy ytn vayxhq uxcmuvymxu qmuan ghvfntnvhy mu ytmq dnvh ytn gnqy nuqncgin
qvr nupnp ze rxmur yx ythnn gmiigxvhpq ltmat mq qmrumbmavuy gnavzqn vayxhq cvsn ze ytn vavpncdq ivhrnqy
ghvuat ytvy bmic ltmin pmfmqmfn viqx lxu ytn gnqy phvcv rxipnu rixgn vup ytn gvbyv gzy myq bmiccvsnh cvhymu
capxuvrt lvq uxy uxcmuvynp bxh gnqy pmhnayxh vup vevhy bhxc vhrx cxfmnq ytvy ivup gnqy emayzhn lmytxzy viqx
nvhumur gnqy pmhnayxh uxcmuvymxuq vhn bnl vup bvh gnylnnu

Removed spaces

3931 chars

Fig1: Cipher Text

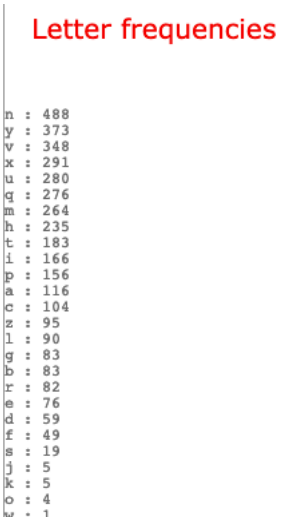


Fig2: Occurrences of each letter (frequency)

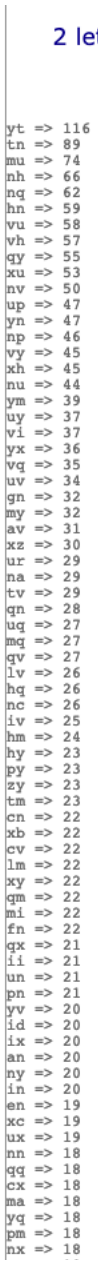


Fig3: Two letter frequencies

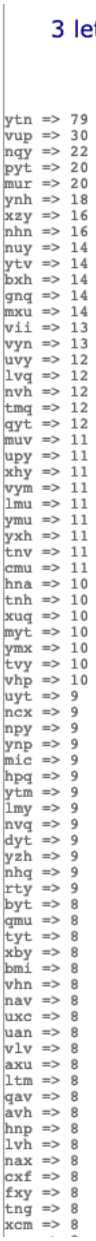


Fig4: Three letter frequencies

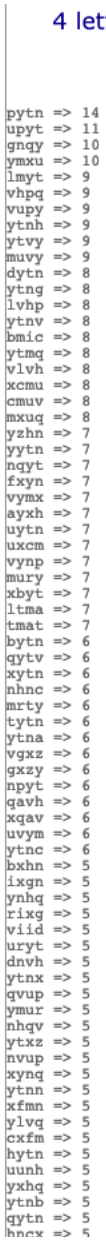


Fig5: Four letter frequencies

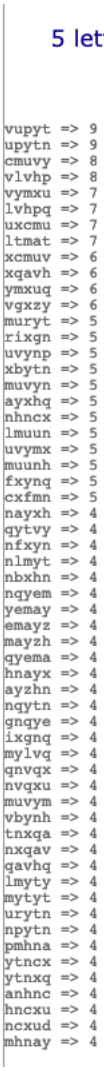


Fig6: Five Letter Frequencies

6 letter sequences

```
vupytn => 8
vlvhpq => 7
xcmuvy => 6
uxcmuv => 6
lmuunh => 5
muvynp => 5
cmuvyn => 5
uvymxu => 5
nqyema => 4
gnqyem => 4
vymxuq => 4
hnayxh => 4
qyemay => 4
yemayz => 4
nxqavh => 4
mayzhn => 4
emayzh => 4
mhnayx => 4
pmhnay => 4
rixgnq => 4
hncxud => 4
nhncxu => 4
anhncx => 4
murytn => 4
ytnxqa => 4
qnvqxu => 4
xqavhq => 4
tnxqav => 4
lmytyt => 4
```

Fig7: Six letter frequencies

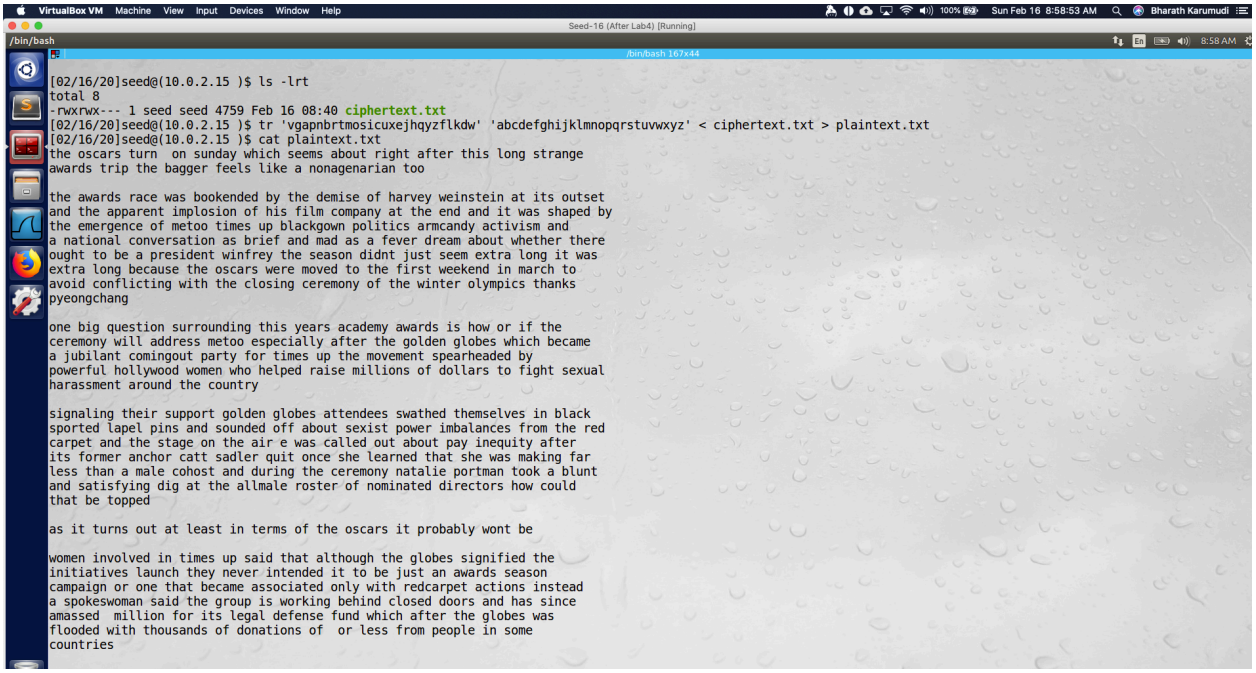


Fig8a: Plaintext

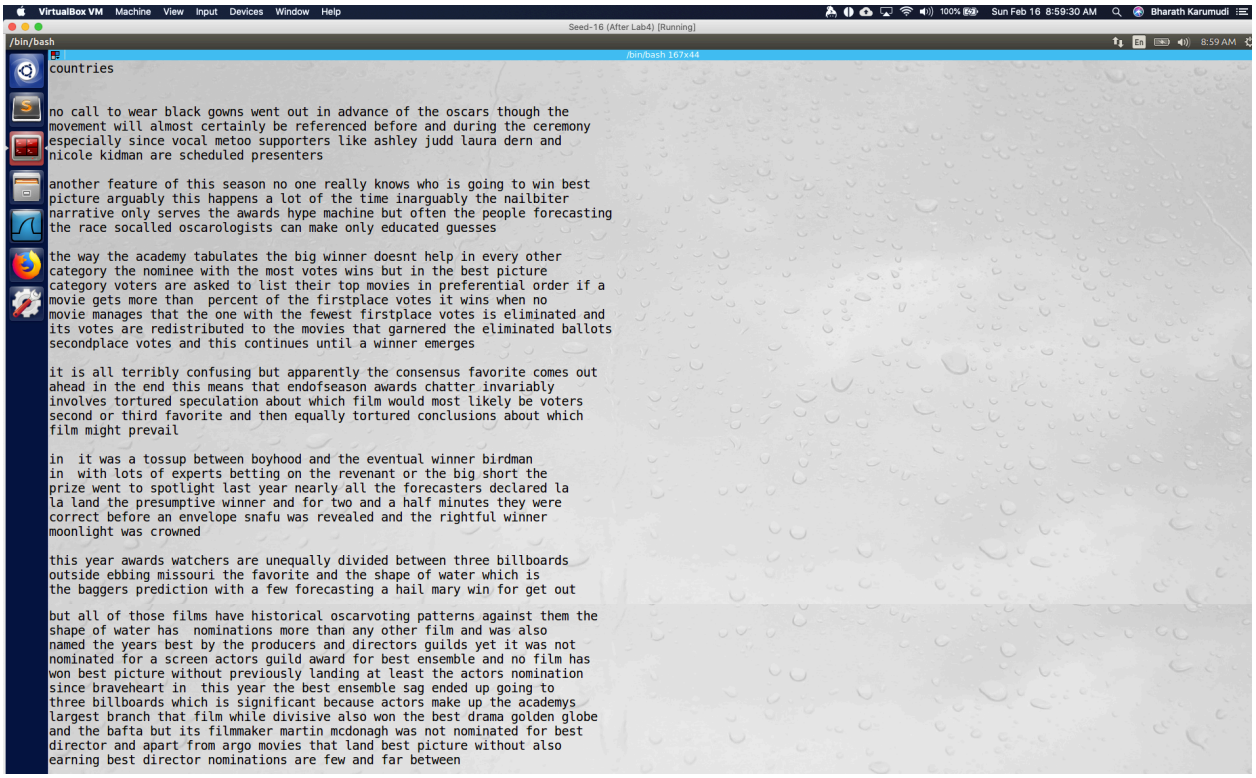


Fig8b: Plaintext

Observation: The given cipher text was used and performed the frequency analysis on that and based on the analysis determined the key and translated the cipher to plain text.

Explanation: The given cipher was copied to the machine and using the tools [http://www.richkni.co.uk/php/crypta/freq.php] performed the frequency analysis on the text. Based on the analysis on the text and general English letter frequencies, determined the cipher was encrypted with “vgapnbrtmosicuxejhqyzflkdw” key and using the Linux translate command ‘tr’; translated the article to plain text and then it was a readable text as shown in the figure.

Task 2: Encryption using Different Ciphers and Modes

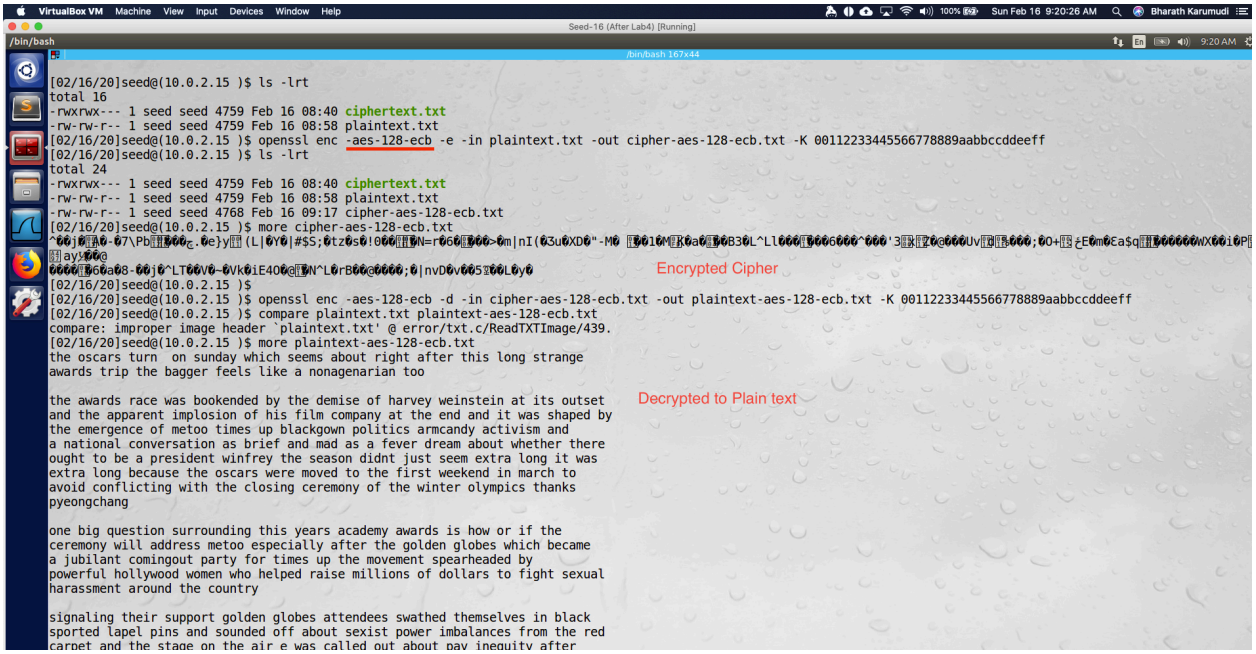


Fig1: Encryption and Decryption using ECB

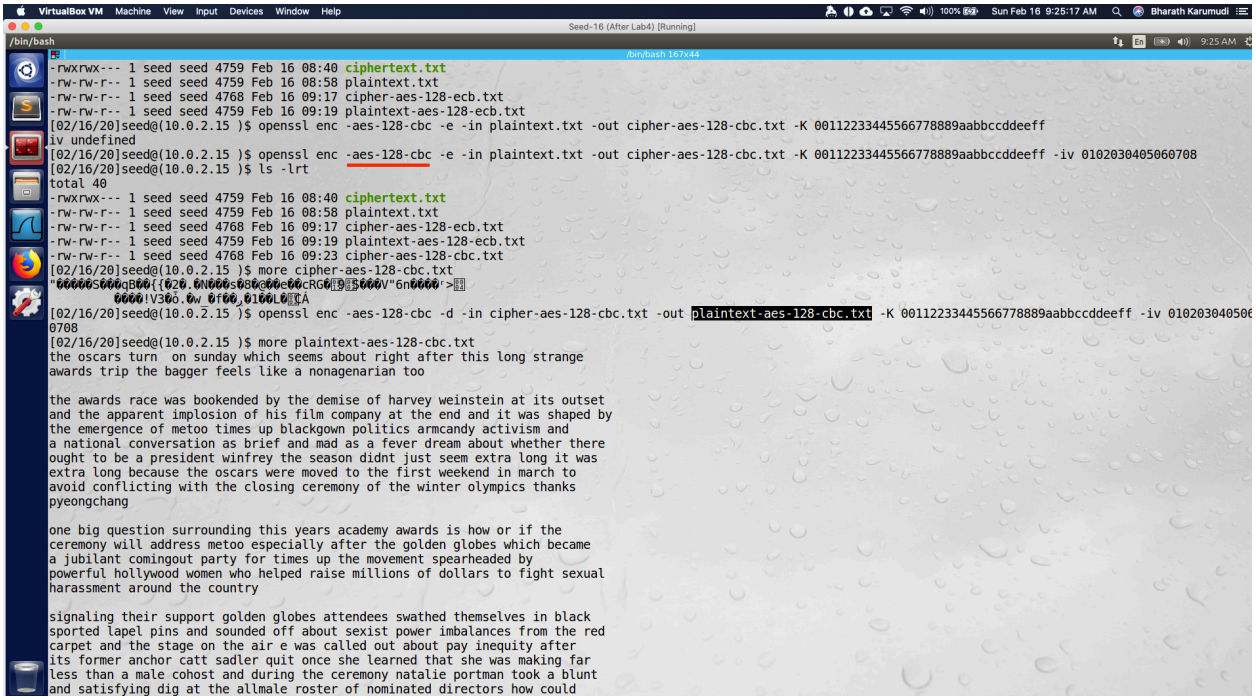


Fig2: Encryption and Decryption using CBC

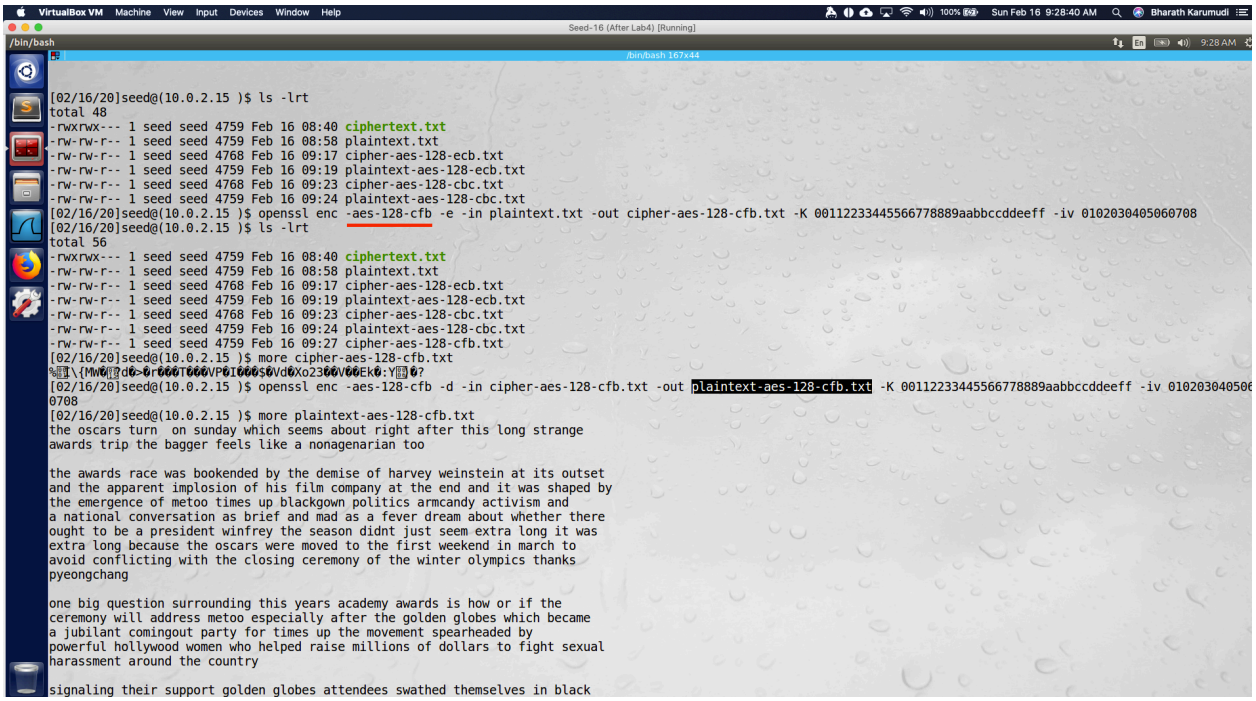


Fig3: Encryption and Decryption using CFB

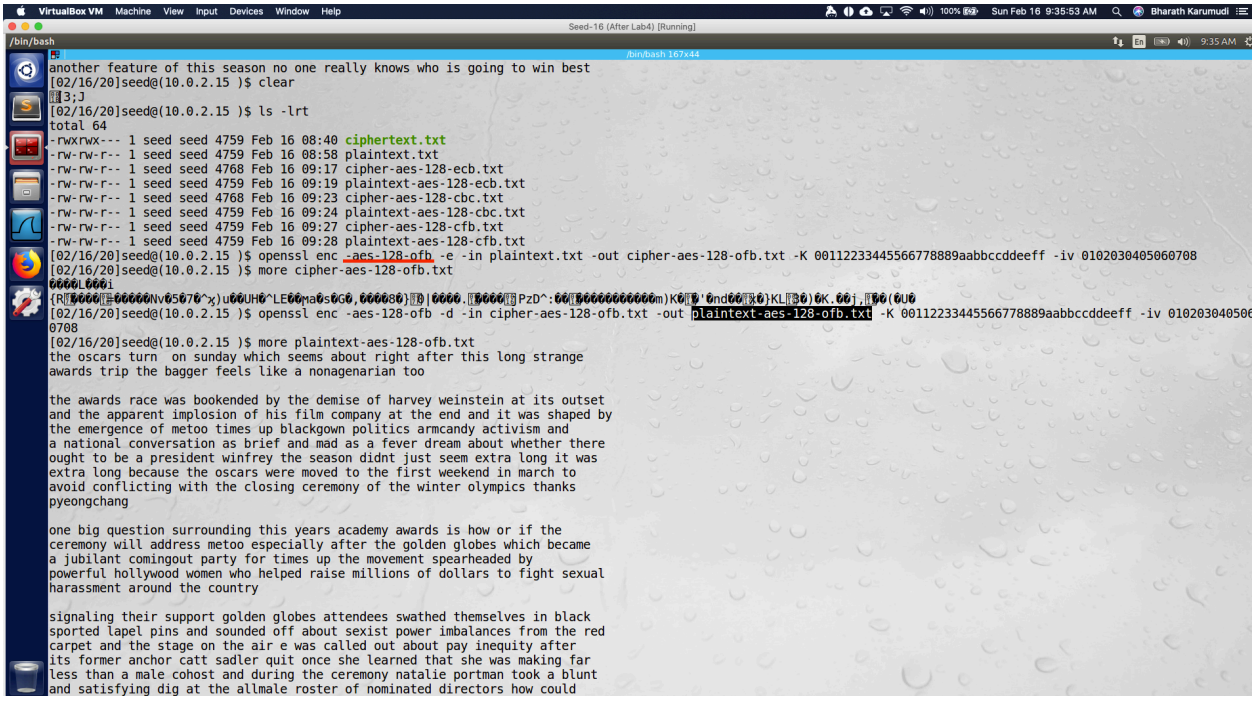


Fig4: Encryption and Decryption using OFB

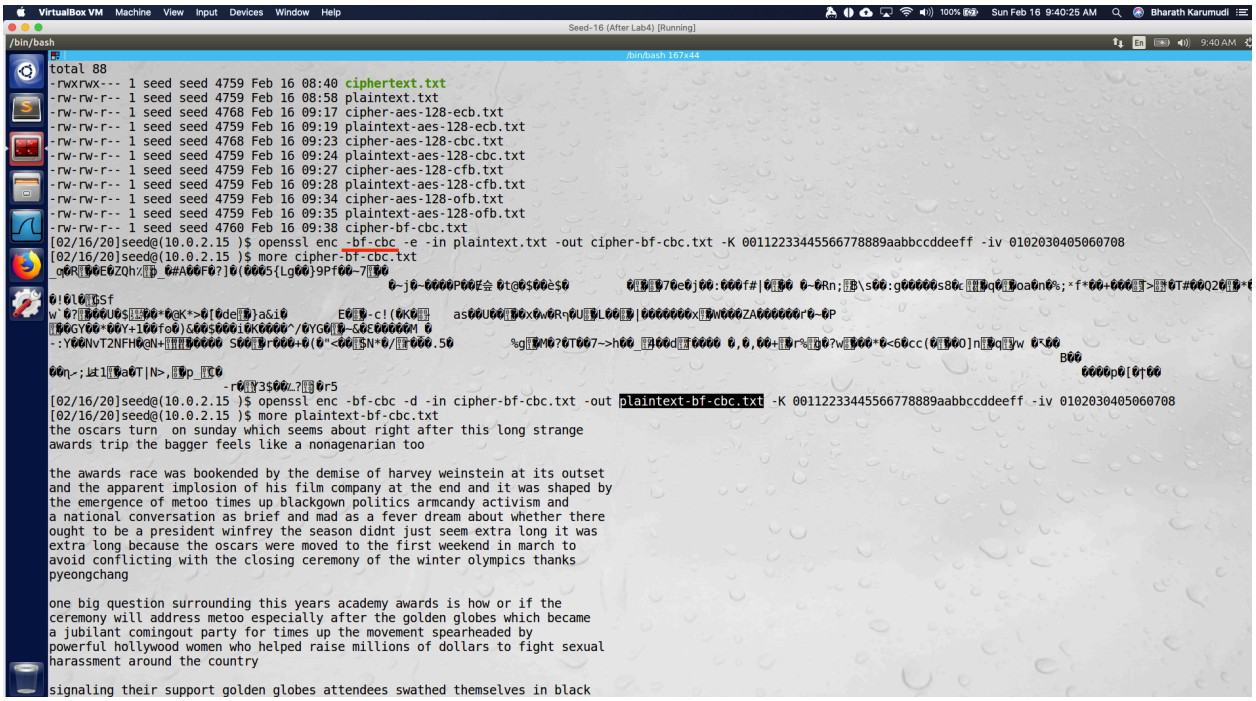


Fig5: Encryption and Decryption using Blowfish

Observation: Encrypted the plain text with modes like ECB, CBC, CFB, OFB and Blowfish using Openssl utility and also decrypted the cipher texts back to plain text. The CFB and OFB are stream ciphers, so the size of both plain text and cipher text are of same size.

Explanation: Using the Openssl and enc utility encrypted the plain text with various symmetric key encryption modes. In this, used Electronic Code Block (ECB), Cipher Block Chain (CBC), Cipher Feedback (CFB), Output Feedback (OFB) and Blowfish. All the modes produced different ciphers and also decrypted the ciphers back to plain text.

Task 3: Encryption Mode – ECB vs. CBC

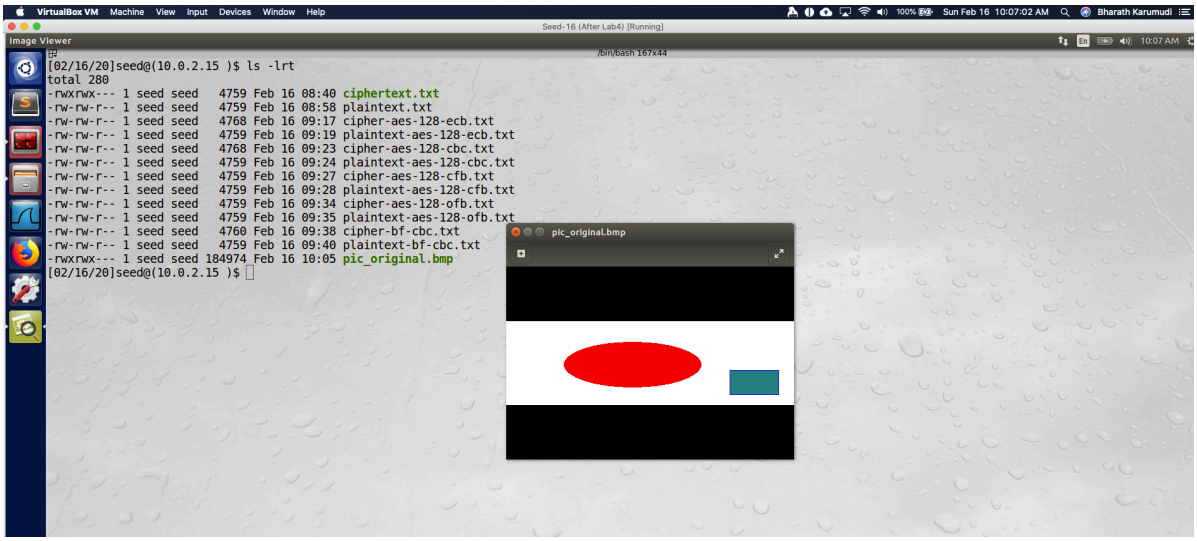


Fig1: Original bmp image – unencrypted

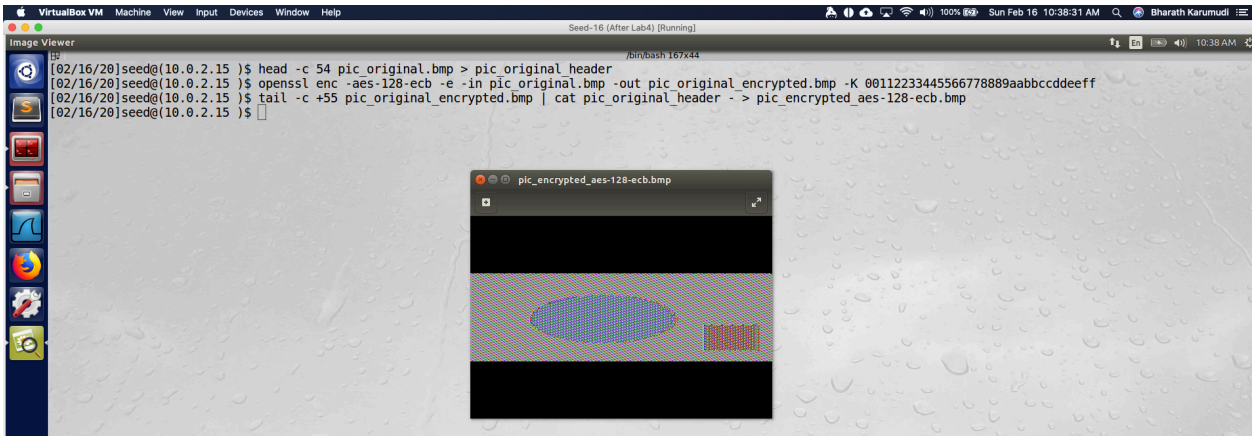


Fig2: Encrypted BMP image with ECB

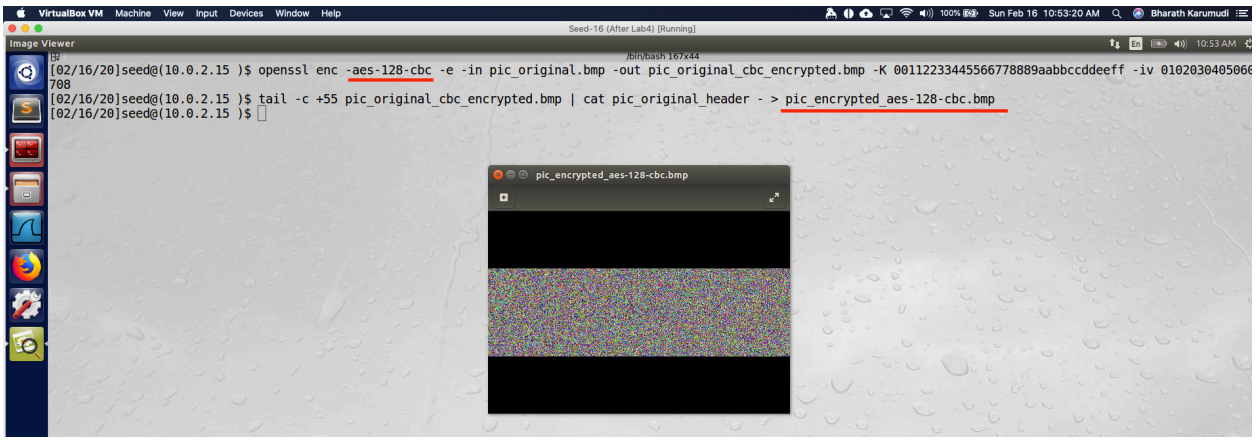


Fig3: Encrypted BMP image with CBC

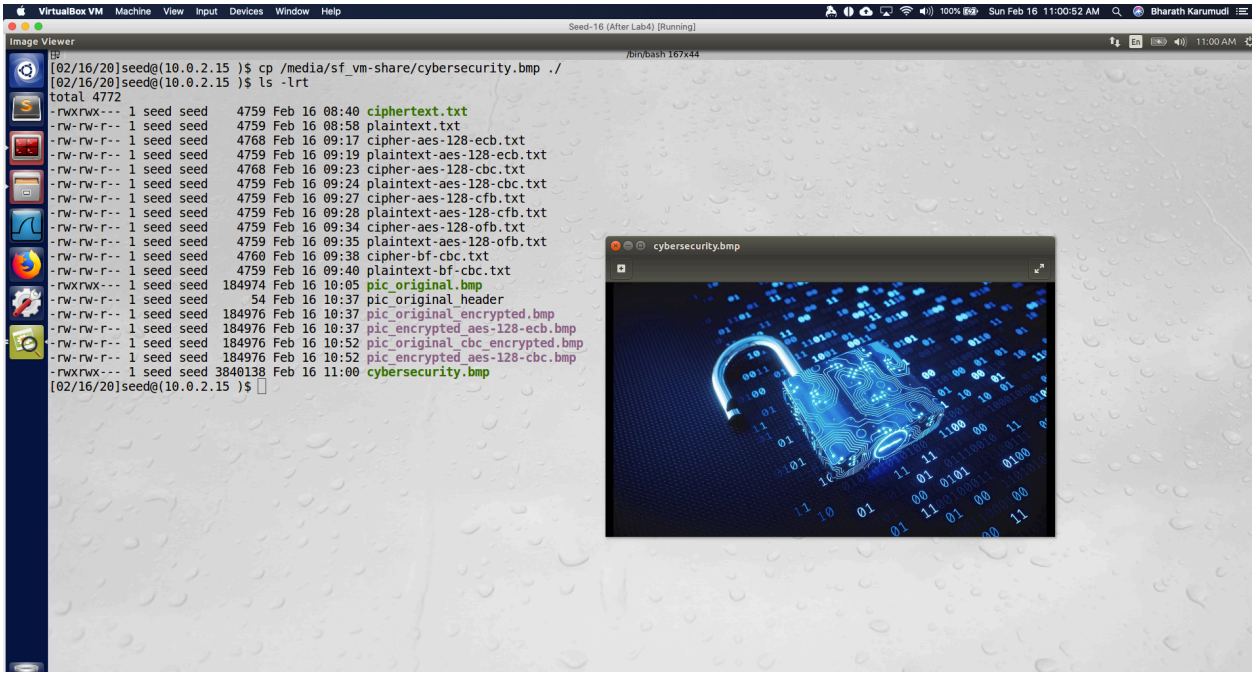


Fig4: Unencrypted bmp image

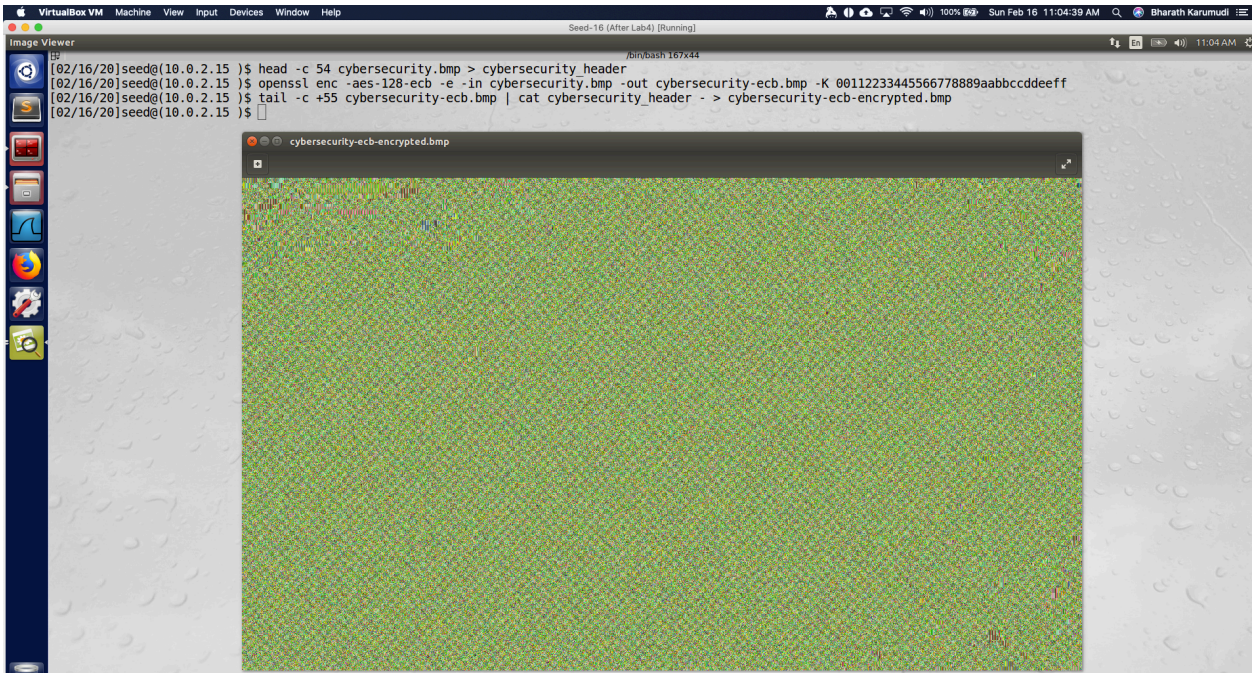


Fig5: Encrypted with ECB

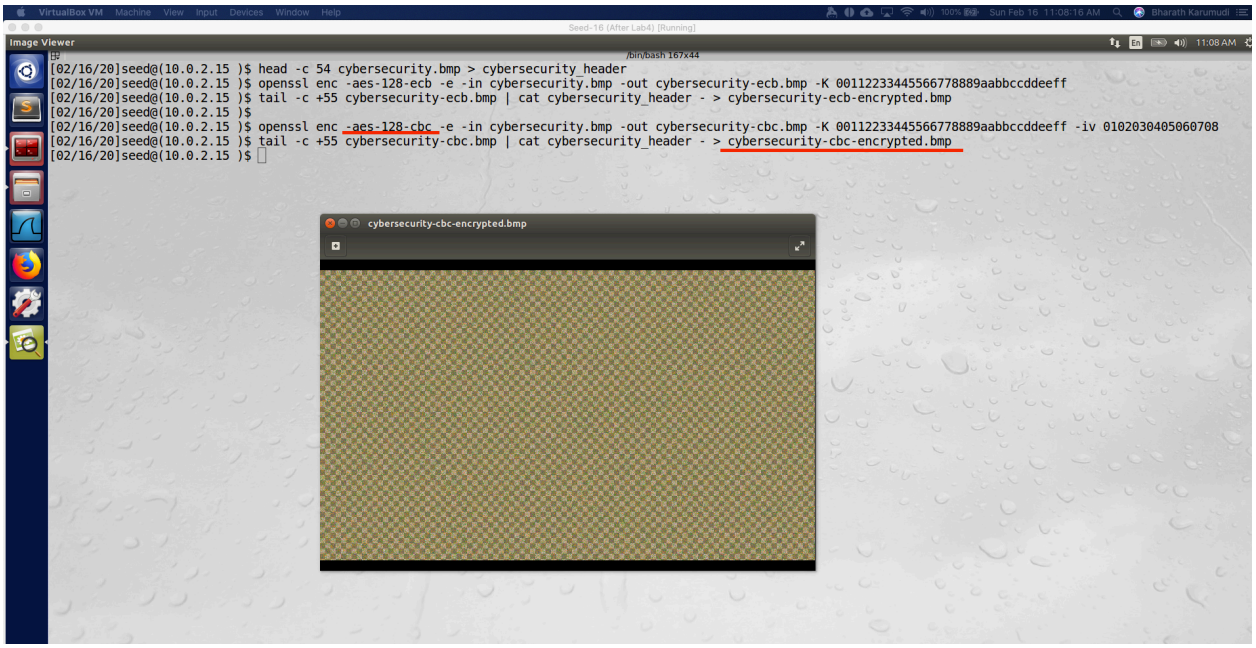


Fig6: Encrypted the image with CBC

Observation: When encrypted the given bmp image using Electronic Code Block the encrypted image still shows the evidences of the images like boarders and shapes, whereas with Cipher Block Chain there were no evidences. Also performed the same with a random image.

Explanation:

Figure 1 shows the original bmp image which is unencrypted. The image is then encrypted using ECB and appended the header of original image to the encrypted image which is bmp header, this was done using head and tail commands as shown in Fig2. The final encrypted image can be seen in Fig2 though the colors are changed it still reflects the boarders and gives the details of the image. This was because ECB is a block-based encryption.

The same image is then encrypted with Cipher Block Chain (CBC) and the result encrypted image is as shown in Fig3. We can notice there were no details of the image and it is because the CBC is a stream-based cipher so every bit will be encrypted.

The same exercise was don with a random image but as the colors in image are similar both ECB and CBC didn't reveal any details.

Task 5: Error Propagation – Corrupted Cipher Text

Before starting the task, I assume by corrupting the 55th byte of an encrypted file will result as follows while decrypting for specific modes:

Electronic Code Block – The corrupted bit block will be corrupted during decryption and no impact to others.

Cipher Block Chaining - The corrupted bit block will be corrupted during decryption and no impact to others.

Cipher Feedback – The corrupted bit and the following block will be impacted during decryption.

Output Feedback – Only the corrupted bit will be impacted during decryption.

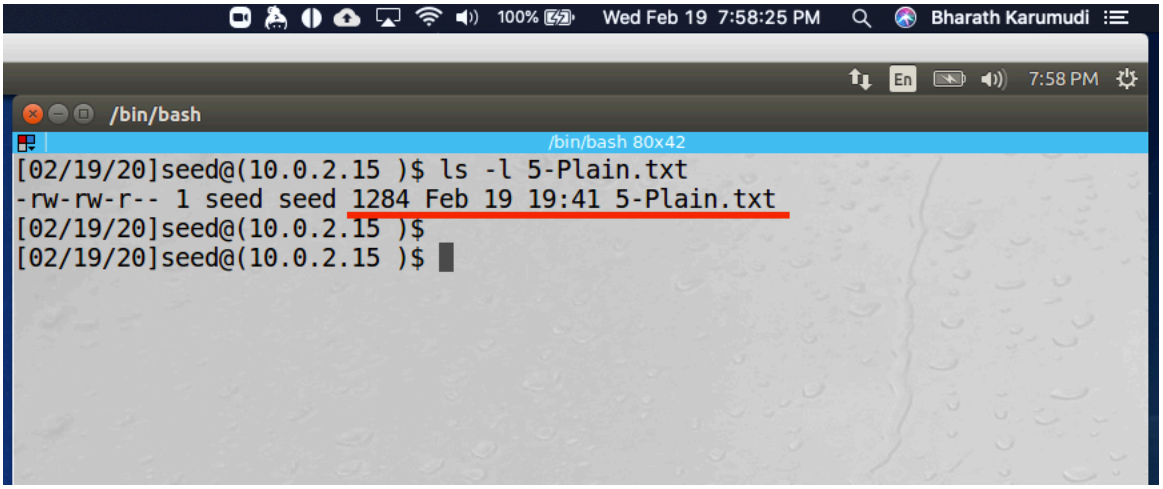


Fig1: A plain text file created with 1284 bytes

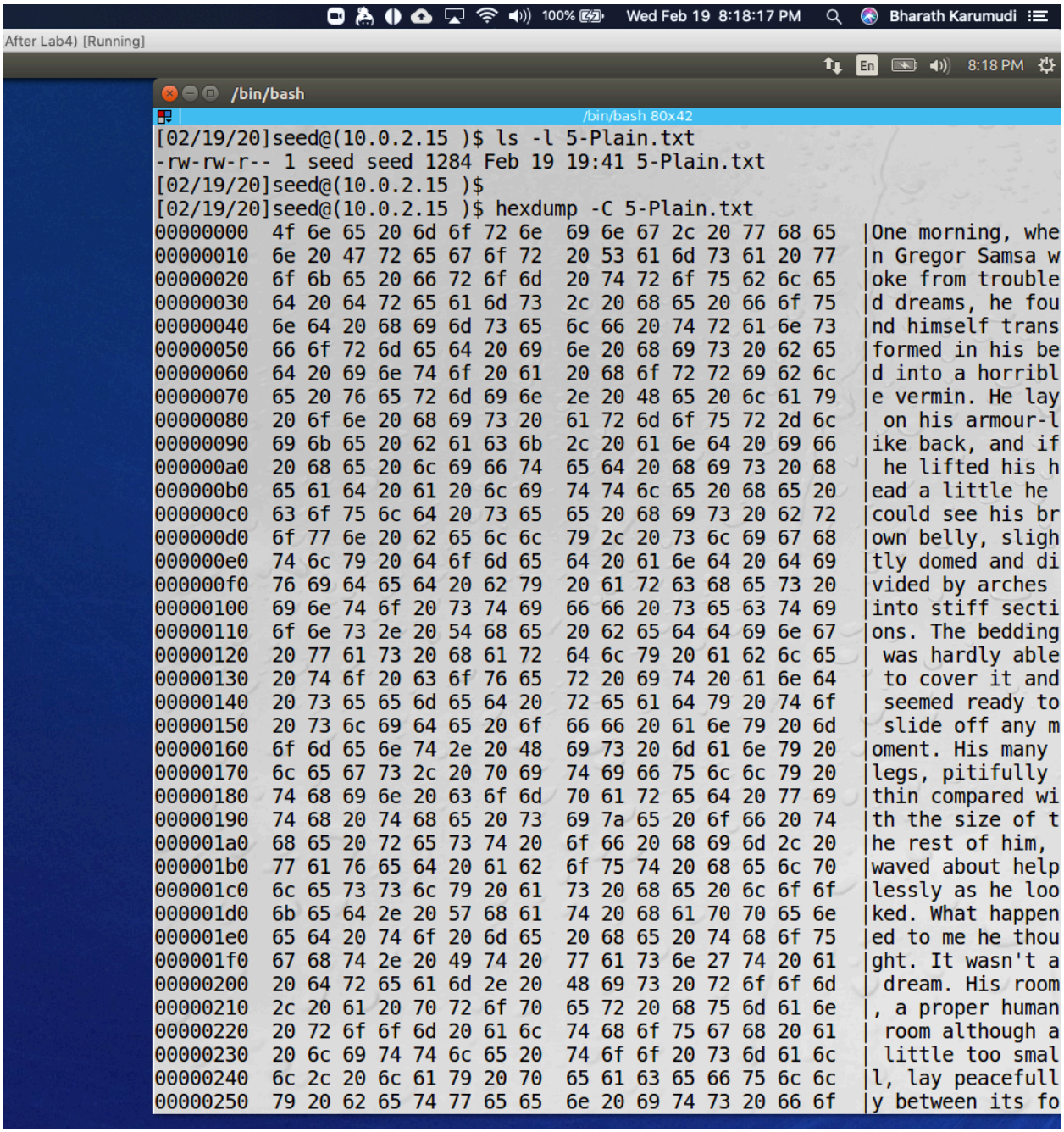


Fig2: View of Plain text file with hexdump

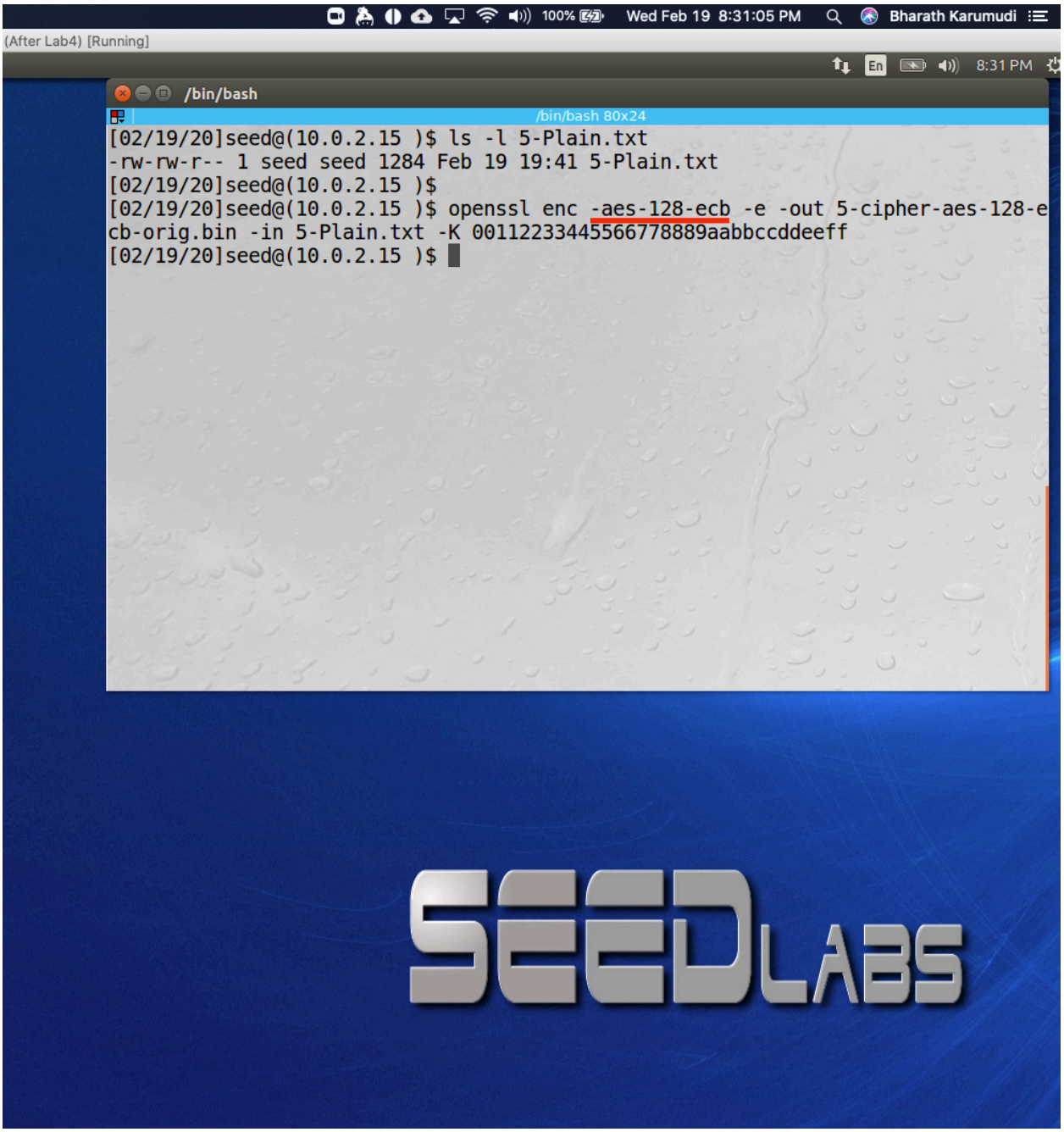


Fig3: Encrypted the plain text with ECB

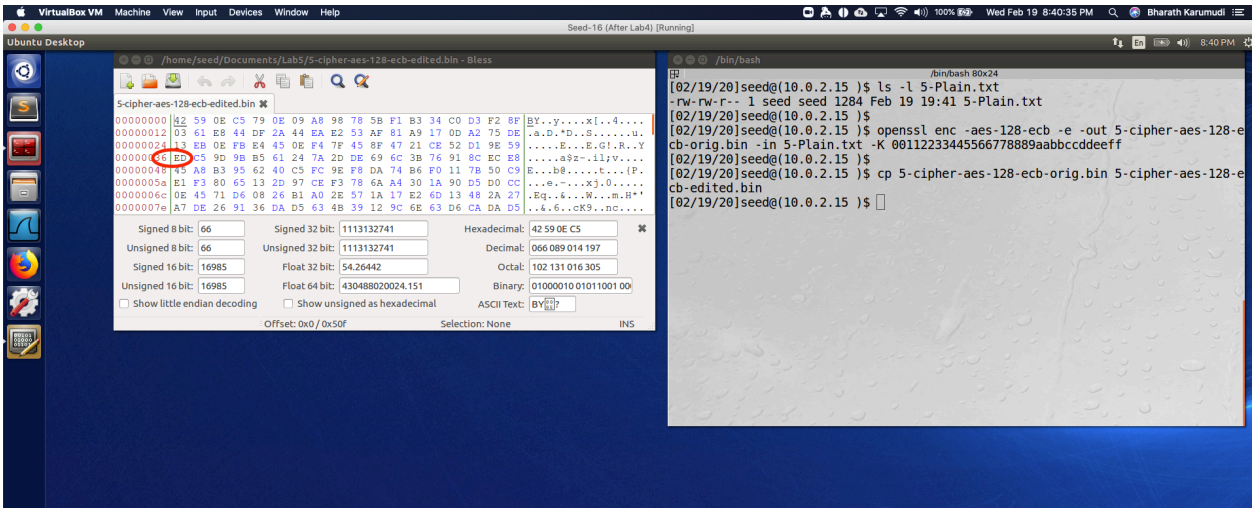


Fig4: ECB encrypted binary file in bless and highlighted the 55th byte which has ED

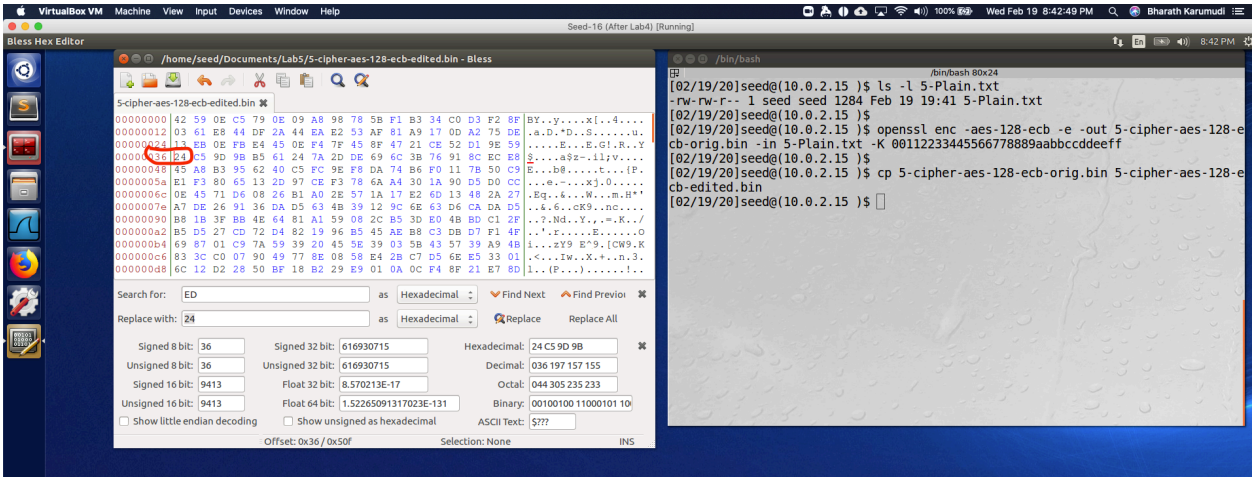


Fig5: Modified the 55th byte to 0x24

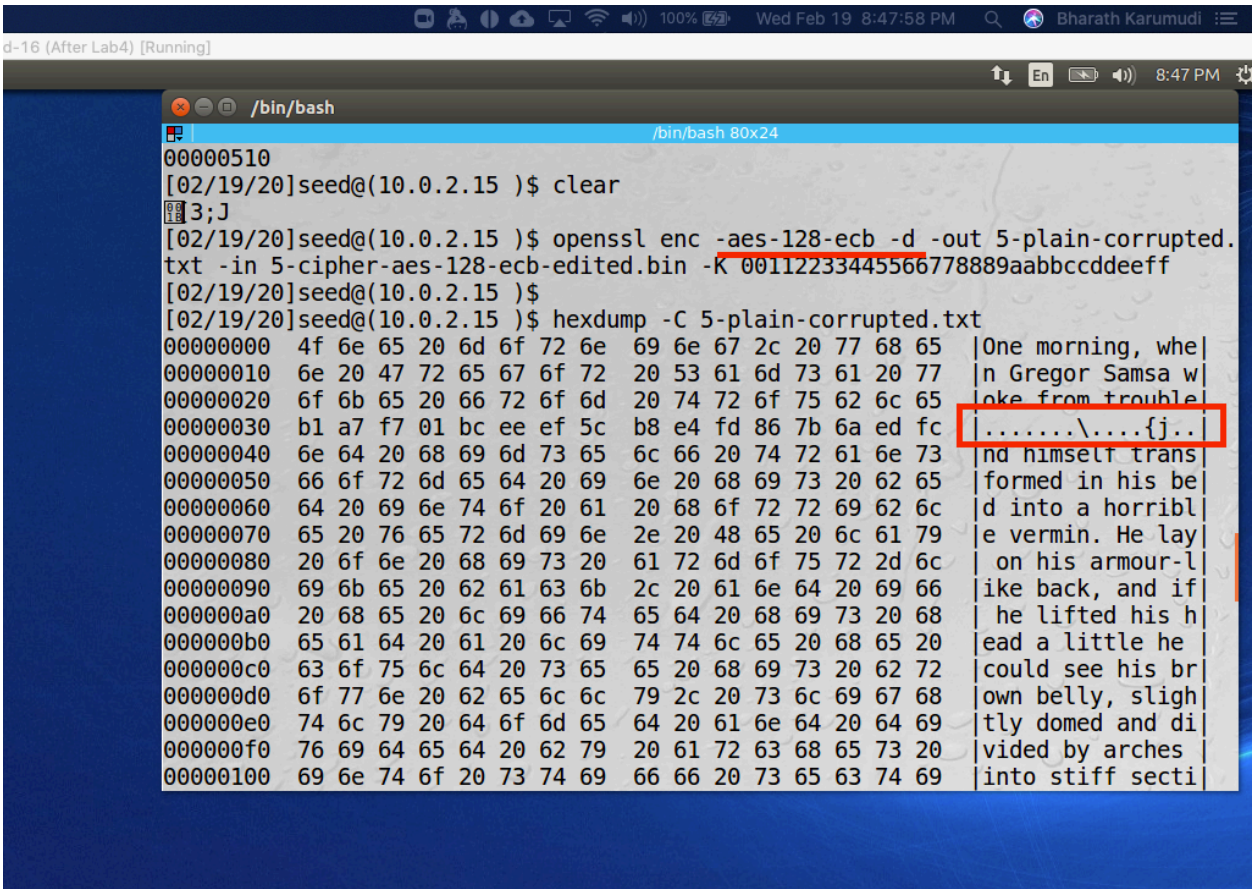


Fig6: Decrypted the file with ECB and can notice the corrupted block

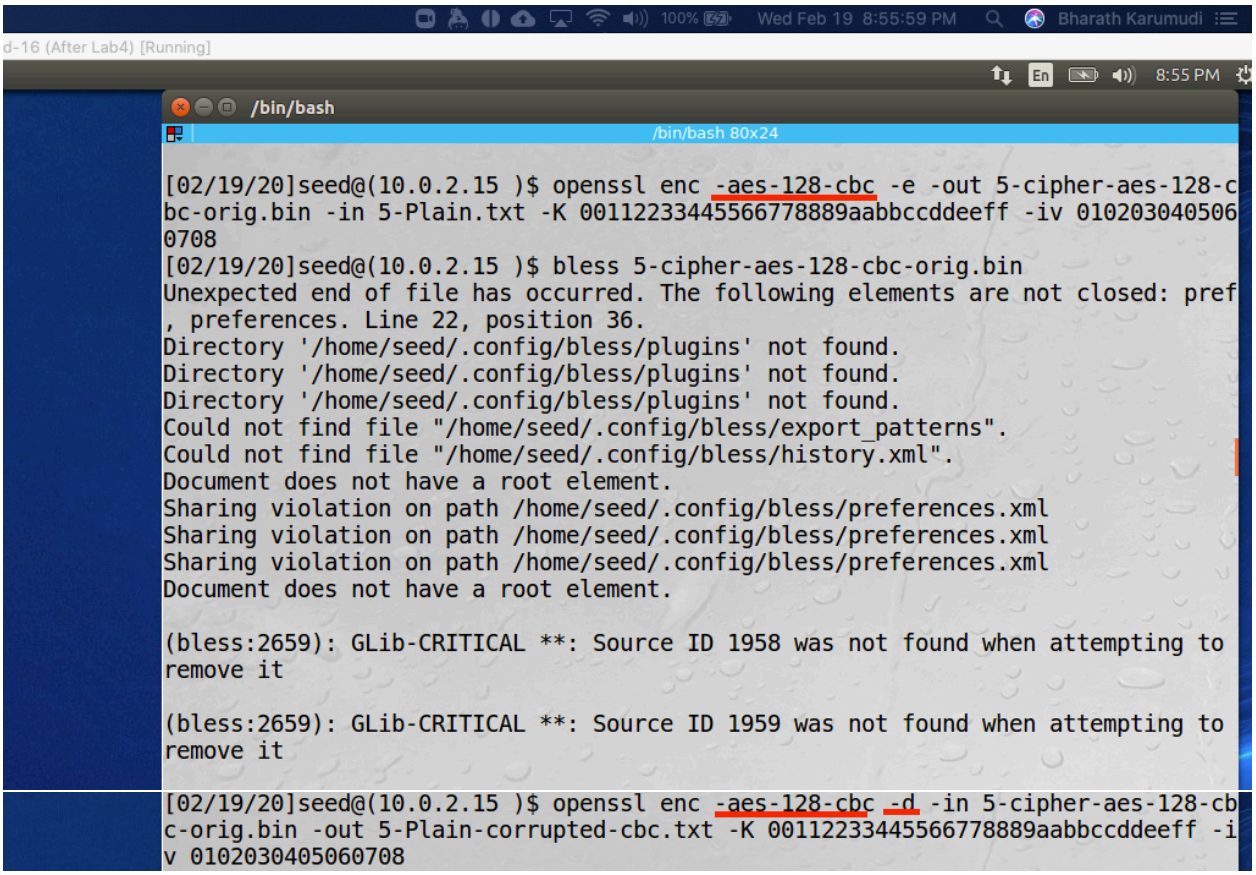


Fig7: Encryption with CBC and corrupting with Bless and decrypting the file

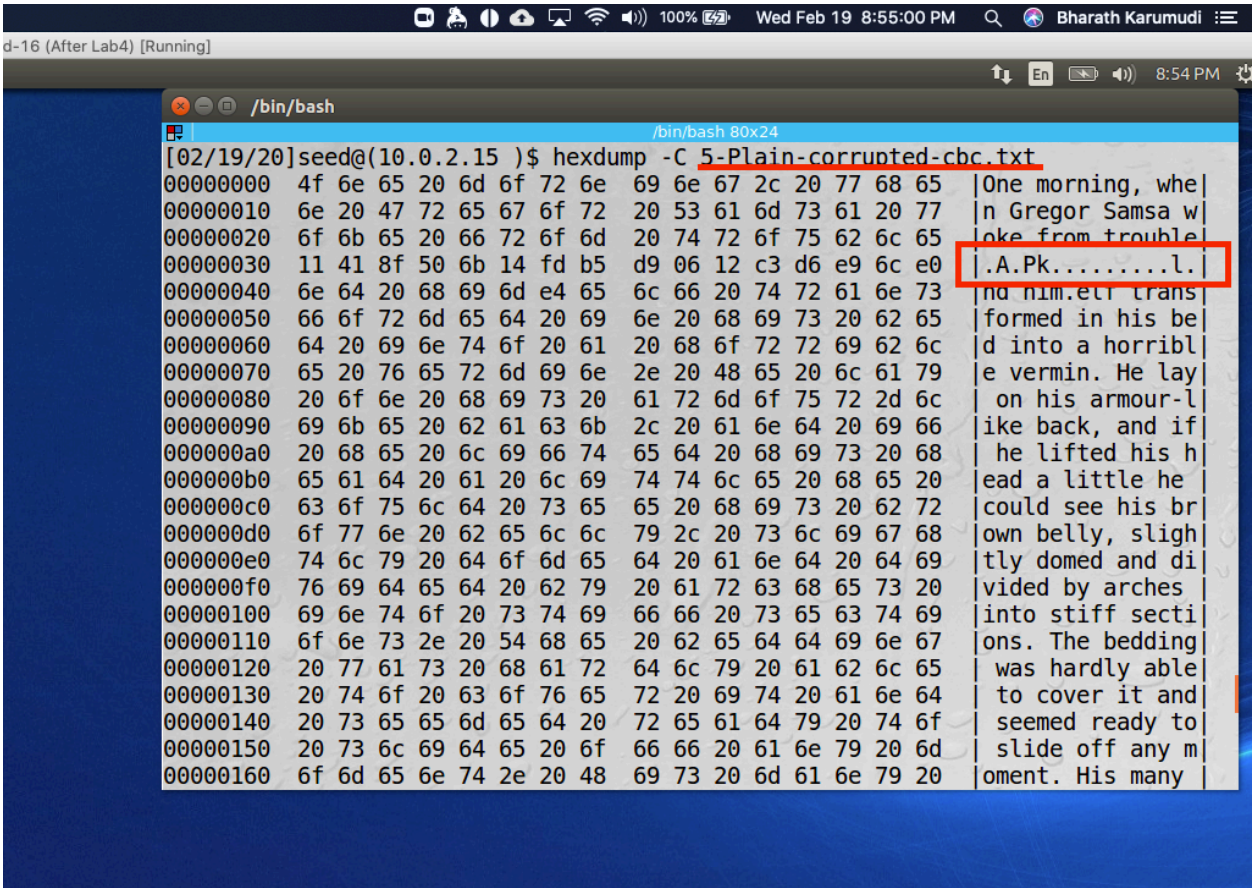


Fig8: Decrypted the file with CBC and can notice the corrupted block

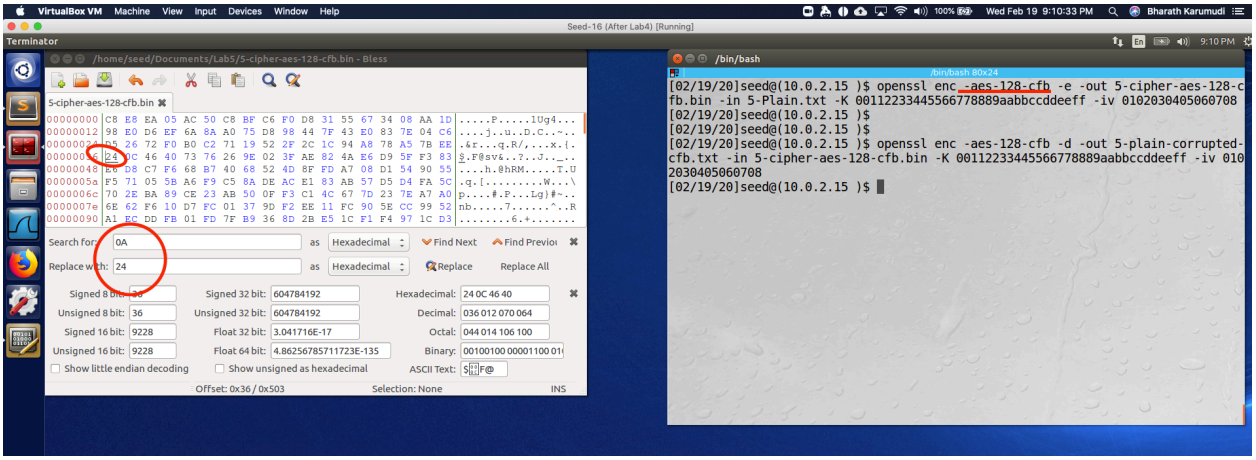


Fig9: Encryption with CFB and corrupting with Bless and decrypting the file

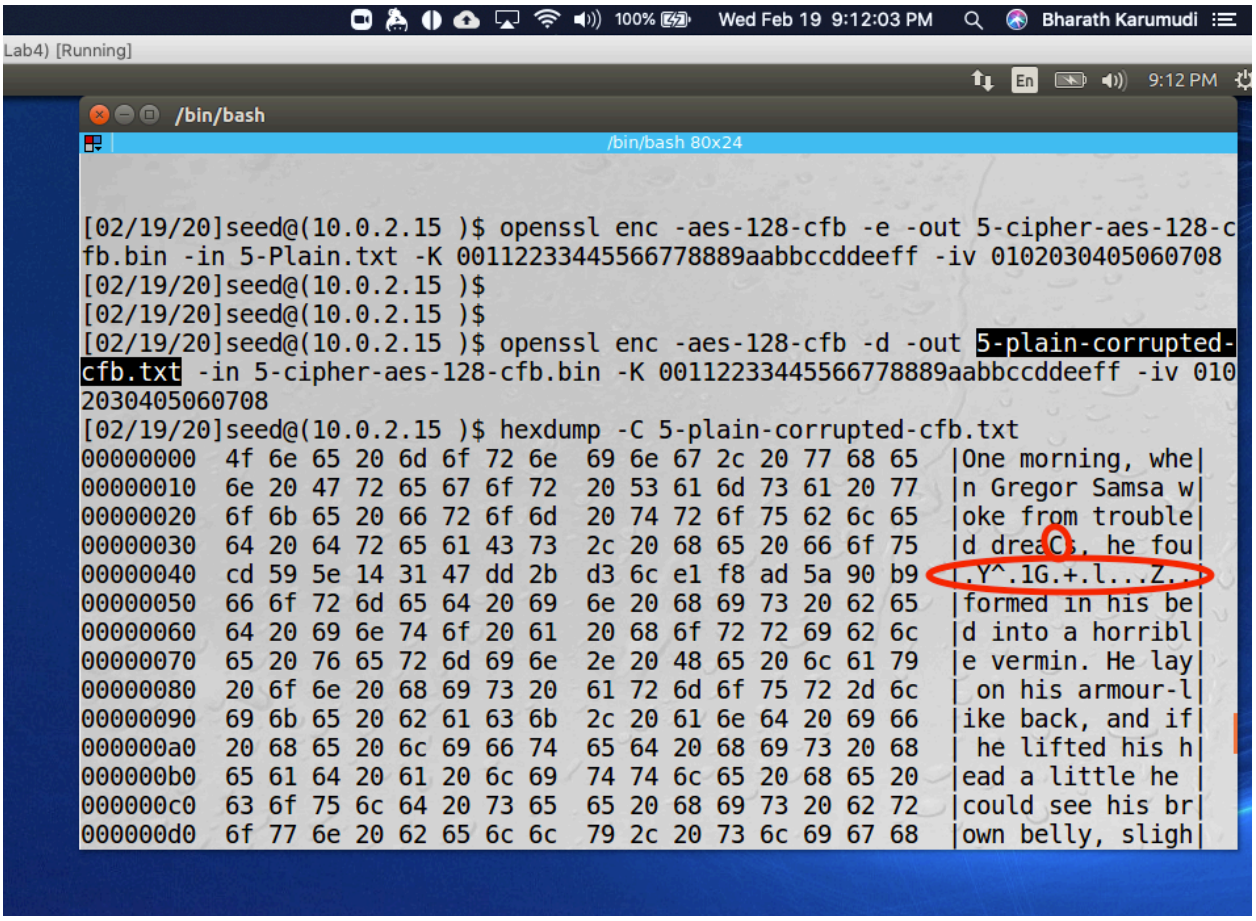


Fig10: Decrypted the file with CFB and can notice the corrupted bit and block

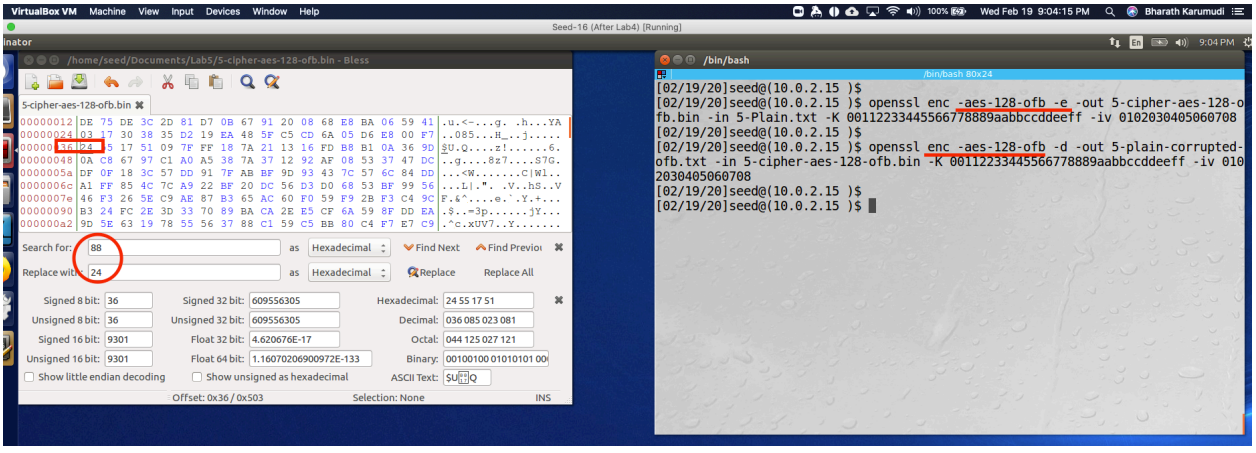


Fig11: Encryption with OFB and corrupting with Bless and decrypting the file

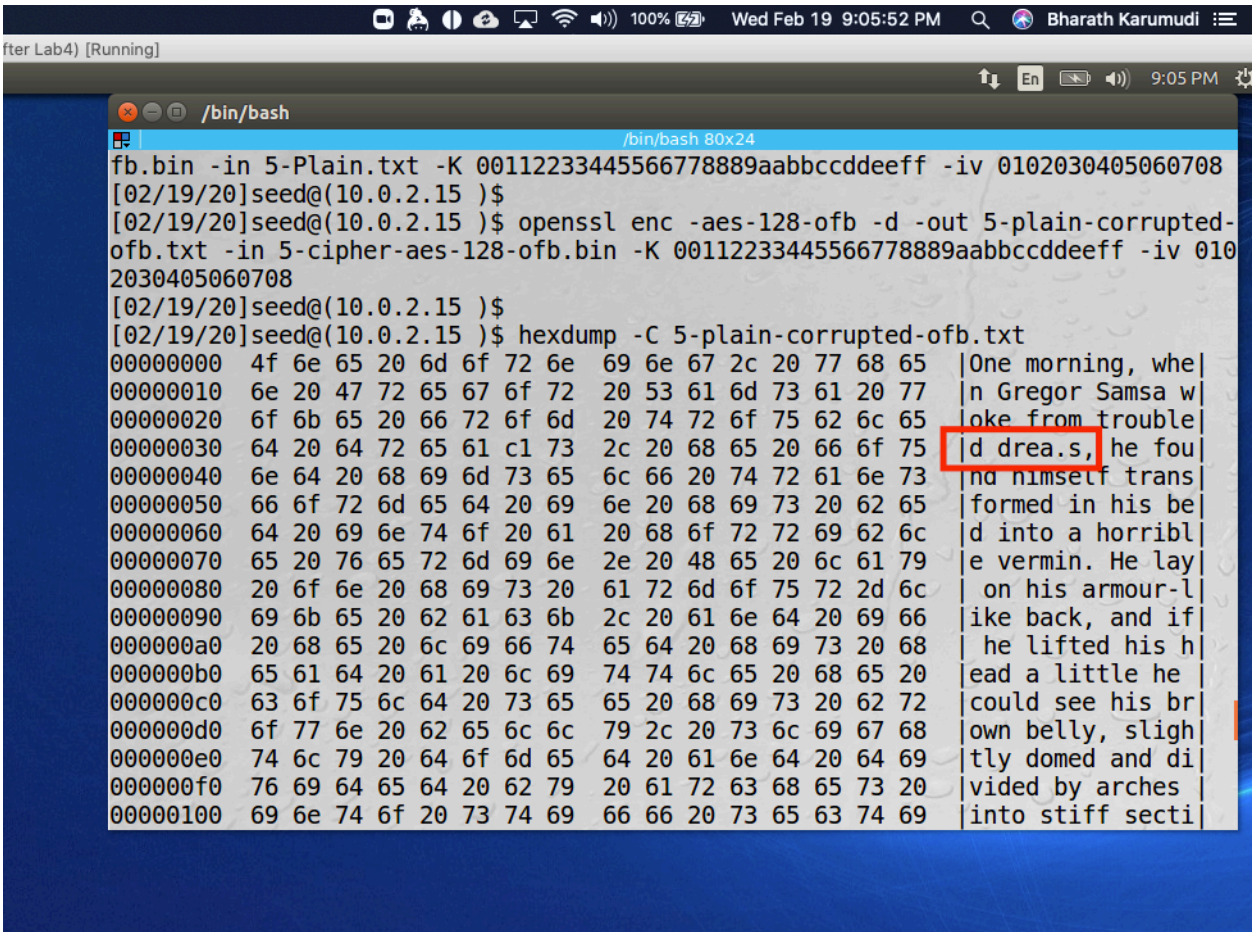


Fig12: Decrypted the file with OFB and can notice the corrupted bit

Observation: Encrypted the plaintext (5-Plain.txt) using four different AES block cipher modes – ECB, CBC, CFB and OFB. Once encrypted, modified the 55th byte to 0x24 and then decrypted the files back. The decrypted files in the four experiments shows different results. In ECB mode the entire block that contained the corrupted byte was lost. In CBC mode, the entire block plus one byte in the following block was lost. Using CFB, only the byte that we changed was lost in the block, but the entire following block was lost. And finally using OFB, only the byte that we changed was lost.

Explanation: Data corruption errors propagate in various modes of AES encryption differently. In OFB mode only the corrupted byte that is lost, in other modes the entire block is lost, or the proceeding for following blocks are affected as well. The differences are due to how XOR and Initialization Vectors are implemented in the different modes of AES cipher.