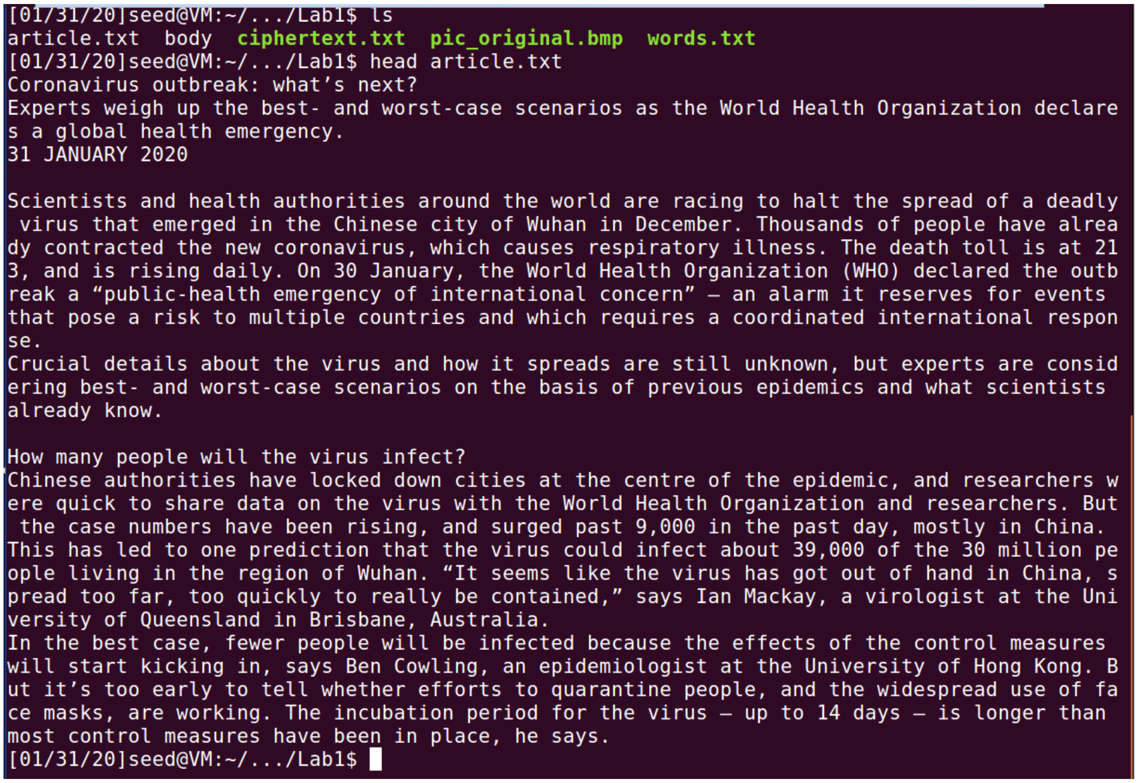
SECRET-KEY ENCRYPTION LAB

Jasmine Joy (500924677)

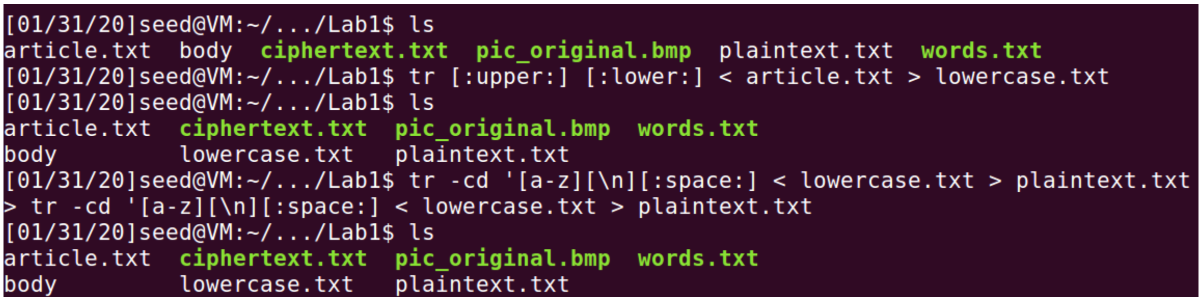
Task 1: Frequency Analysis Against Monoalphabetic Substitution Cipher

(Step 1) The article used for encryption in this task is as below:



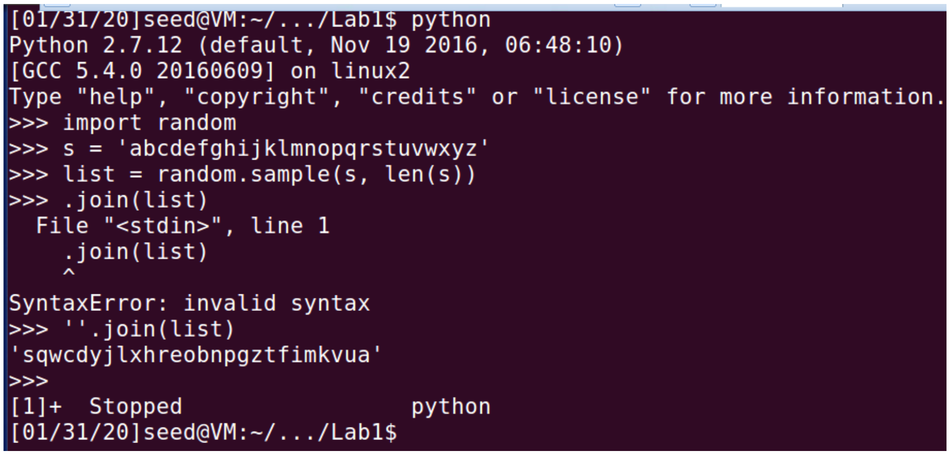
```
[01/31/20]seed@VM:~/.../Lab1$ ls
article.txt  body  ciphertext.txt  pic_original.bmp  words.txt
[01/31/20]seed@VM:~/.../Lab1$ head article.txt
Coronavirus outbreak: what’s next?
Experts weigh up the best- and worst-case scenarios as the World Health Organization declare
s a global health emergency.
31 JANUARY 2020

Scientists and health authorities around the world are racing to halt the spread of a deadly
 virus that emerged in the Chinese city of Wuhan in December. Thousands of people have alrea
dy contracted the new coronavirus, which causes respiratory illness. The death toll is at 21
3, and is rising daily. On 30 January, the World Health Organization (WHO) declared the outb
reak a “public-health emergency of international concern” — an alarm it reserves for events
that pose a risk to multiple countries and which requires a coordinated international respon
se.
Crucial details about the virus and how it spreads are still unknown, but experts are consid
ering best- and worst-case scenarios on the basis of previous epidemics and what scientists
already know.

How many people will the virus infect?
Chinese authorities have locked down cities at the centre of the epidemic, and researchers w
ere quick to share data on the virus with the World Health Organization and researchers. But
 the case numbers have been rising, and surged past 9,000 in the past day, mostly in China.
This has led to one prediction that the virus could infect about 39,000 of the 30 million pe
ople living in the region of Wuhan. “It seems like the virus has got out of hand in China, s
pread too far, too quickly to really be contained,” says Ian Mackay, a virologist at the Uni
versity of Queensland in Brisbane, Australia.
In the best case, fewer people will be infected because the effects of the control measures
will start kicking in, says Ben Cowling, an epidemiologist at the University of Hong Kong. B
ut it’s too early to tell whether efforts to quarantine people, and the widespread use of fa
ce masks, are working. The incubation period for the virus — up to 14 days — is longer than
most control measures have been in place, he says.
[01/31/20]seed@VM:~/.../Lab1$ █
```

Below is the conversion of the article into lowercase and then into plaintext (removing spaces):



```
[01/31/20]seed@VM:~/.../Lab1$ ls
article.txt  body  ciphertext.txt  pic_original.bmp  plaintext.txt  words.txt
[01/31/20]seed@VM:~/.../Lab1$ tr [:upper:] [:lower:] < article.txt > lowercase.txt
[01/31/20]seed@VM:~/.../Lab1$ ls
article.txt  ciphertext.txt  pic_original.bmp  words.txt
body         lowercase.txt   plaintext.txt
[01/31/20]seed@VM:~/.../Lab1$ tr -cd '[a-z][\n][:space:] < lowercase.txt > plaintext.txt
> tr -cd '[a-z][\n][:space:] < lowercase.txt > plaintext.txt
[01/31/20]seed@VM:~/.../Lab1$ ls
article.txt  ciphertext.txt  pic_original.bmp  words.txt
body         lowercase.txt   plaintext.txt
```

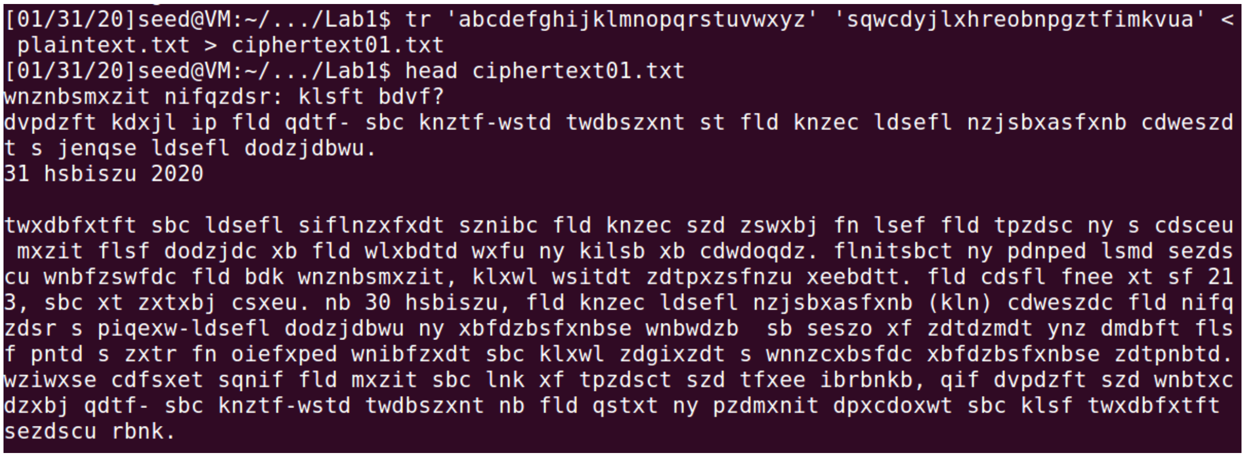
(Step 2) The following python code was used to generate the encryption key:



```
[01/31/20]seed@VM:~/.../Lab1$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import random
>>> s = 'abcdefghijklmnopqrstuvwxyz'
>>> list = random.sample(s, len(s))
>>> .join(list)
  File "<stdin>", line 1
    .join(list)
    ^
SyntaxError: invalid syntax
>>> ''.join(list)
'sqwcdyjlxhreobnpgztfimkvua'
>>>
[1]+  Stopped                 python
[01/31/20]seed@VM:~/.../Lab1$
```

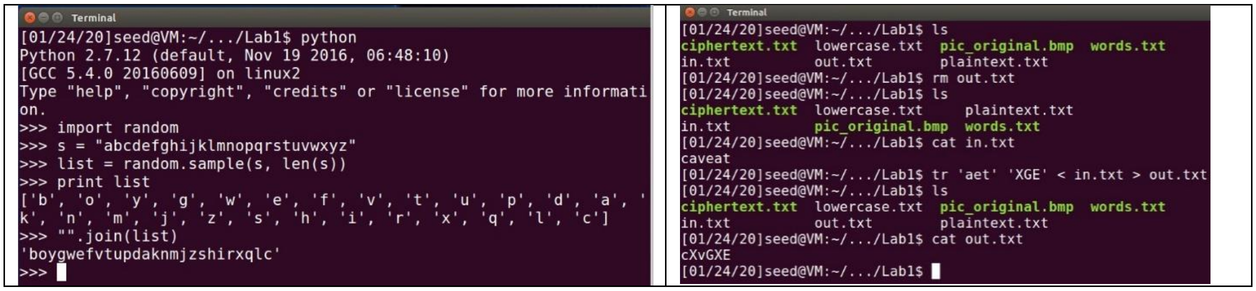
The encryption key: ‘*sqwcdyjlxhreobnpgztfimkvua*’

(Step 3) Encryption:



```
[01/31/20]seed@VM:~/.../Lab1$ tr 'abcdefghijklmnopqrstuvwxyz' 'sqwcdyjlxhreobnpgztfimkvua' <
 plaintext.txt > ciphertext01.txt
[01/31/20]seed@VM:~/.../Lab1$ head ciphertext01.txt
wnznbsmxzit nifqzdsr: klsft bdvf?
dvpdzft kdxjl ip fld qdtf- sbc knztf-wstd twdbszxnt st fld knzec ldsefl nzjsbxasfxnb cdweszd
t s jenqse ldsefl dodzjdbwu.
31 hsbiszu 2020

twxdbfxtft sbc ldsefl siflnzxfxdt sznibc fld knzec szd zswxbj fn lsef fld tpzdsc ny s cdsceu
 mxzit flsf dodzjdc xb fld wlxbdtd wxfu ny kilsb xb cdwdoqdz. flnitsbct ny pdnped lsmd sezds
cu wnbfzswfdc fld bdk wnznbsmxzit, klxwl wsitdt zdtpxzsfnzu xeebdtt. fld cdsfl fnee xt sf 21
3, sbc xt zxtxbj csxeu. nb 30 hsbiszu, fld knzec ldsefl nzjsbxasfxnb (kln) cdweszdc fld nifq
zdsr s piqexw-ldsefl dodzjdbwu ny xbfdzbsfxnbse wnbwdzb  sb seszo xf zdtdzmdt ynz dmdbft fls
f pntd s zxtr fn oiefxped wnibfzxdt sbc klxwl zdgixzdt s wnnzcxbsfdc xbfdzbsfxnbse zdtpnbtd.
wziwxse cdfsxet sqnif fld mxzit sbc lnk xf tpzdsct szd tfxee ibrbnkb, qif dvpdzft szd wnbtxc
dzxbj qdtf- sbc knztf-wstd twdbszxnt nb fld qstxt ny pzdmxnit dpxcdoxwt sbc klsf twxdbfxtft
sezdscu rbnk.
```

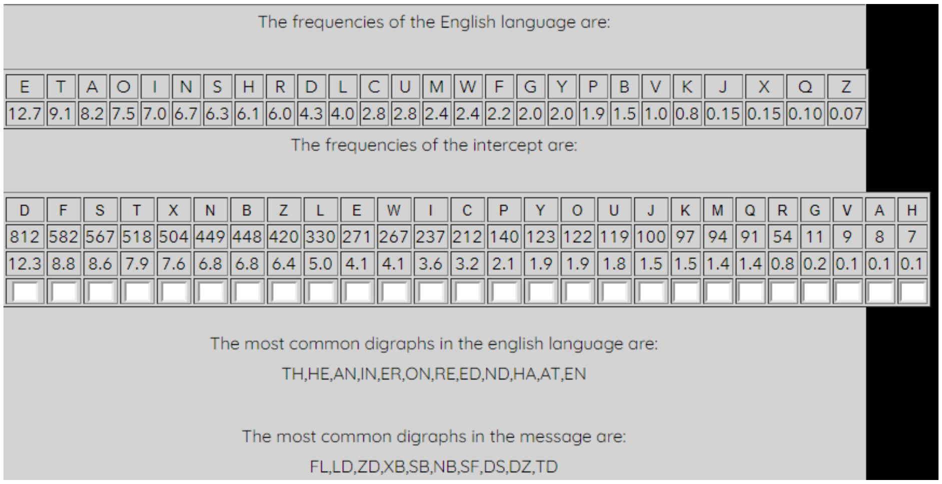
Creating one more ciphertext using plaintext and encryption key:



```
[01/24/20]seed@VM:~/.../Lab1$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more informati
on.
>>> import random
>>> s = "abcdefghijklmnopqrstuvwxyz"
>>> list = random.sample(s, len(s))
>>> print list
['b', 'o', 'y', 'g', 'w', 'e', 'f', 'v', 't', 'u', 'p', 'd', 'a', '
k', 'n', 'm', 'j', 'z', 's', 'h', 'i', 'r', 'x', 'q', 'l', 'c']
>>> "".join(list)
'boygwefvtupdaknmjzshirxqlc'
>>> 
```

```
[01/24/20]seed@VM:~/.../Lab1$ ls
ciphertext.txt  lowercase.txt  pic_original.bmp  words.txt
in.txt          out.txt        plaintext.txt
[01/24/20]seed@VM:~/.../Lab1$ rm out.txt
[01/24/20]seed@VM:~/.../Lab1$ ls
ciphertext.txt  lowercase.txt     plaintext.txt
in.txt          pic_original.bmp  words.txt
[01/24/20]seed@VM:~/.../Lab1$ cat in.txt
caveat
[01/24/20]seed@VM:~/.../Lab1$ tr 'aet' 'XGE' < in.txt > out.txt
[01/24/20]seed@VM:~/.../Lab1$ ls
ciphertext.txt  lowercase.txt  pic_original.bmp  words.txt
in.txt          out.txt        plaintext.txt
[01/24/20]seed@VM:~/.../Lab1$ cat out.txt
cXvGXE
[01/24/20]seed@VM:~/.../Lab1$ 
```

I used the tool available on https://crypto.interactive-maths.com/frequency-analysis-breaking-the-code.html to perform the frequency analysis:



The frequencies of the English language are:

E	T	A	O	I	N	S	H	R	D	L	C	U	M	W	F	G	Y	P	B	V	K	J	X	Q	Z
12.7	9.1	8.2	7.5	7.0	6.7	6.3	6.1	6.0	4.3	4.0	2.8	2.8	2.4	2.4	2.2	2.0	2.0	1.9	1.5	1.0	0.8	0.15	0.15	0.10	0.07

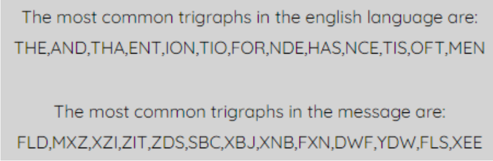
The frequencies of the intercept are:

D	F	S	T	X	N	B	Z	L	E	W	I	C	P	Y	O	U	J	K	M	Q	R	G	V	A	H
812	582	567	518	504	449	448	420	330	271	267	237	212	140	123	122	119	100	97	94	91	54	11	9	8	7
12.3	8.8	8.6	7.9	7.6	6.8	6.8	6.4	5.0	4.1	4.1	3.6	3.2	2.1	1.9	1.9	1.8	1.5	1.5	1.4	1.4	0.8	0.2	0.1	0.1	0.1
□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

The most common digraphs in the english language are:
TH,HE,AN,IN,ER,ON,RE,ED,ND,HA,AT,EN

The most common digraphs in the message are:
FL,LD,ZD,XB,SB,NB,SF,DS,DZ,TD

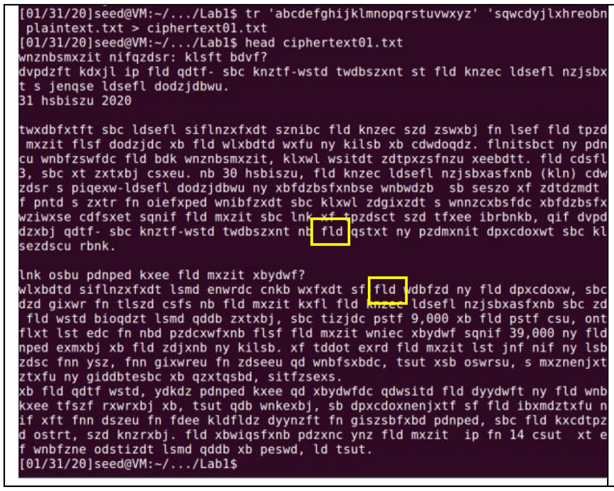
Bigram and Trigram frequencies:



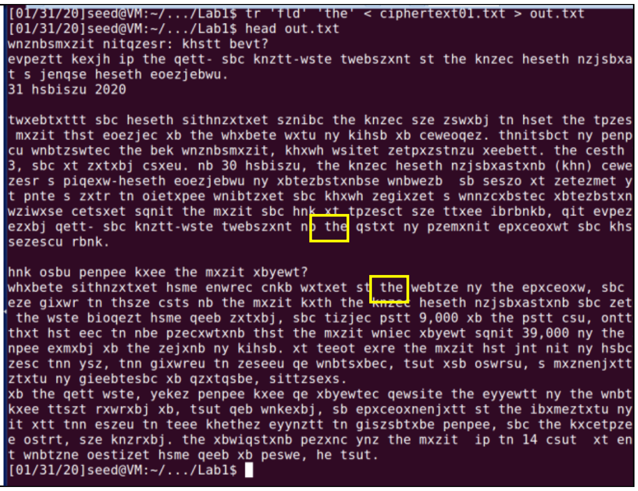
The most common trigraphs in the english language are:
THE,AND,THA,ENT,ION,TIO,FOR,NDE,HAS,NCE,TIS,OFT,MEN

The most common trigraphs in the message are:
FLD,MXZ,XZI,ZIT,ZDS,SBC,XBJ,XNB,FXN,DWF,YDW,FLS,XEE

Using the above information (trigrams), I began replacing ‘fld’ with ‘the’; and the results are as below:



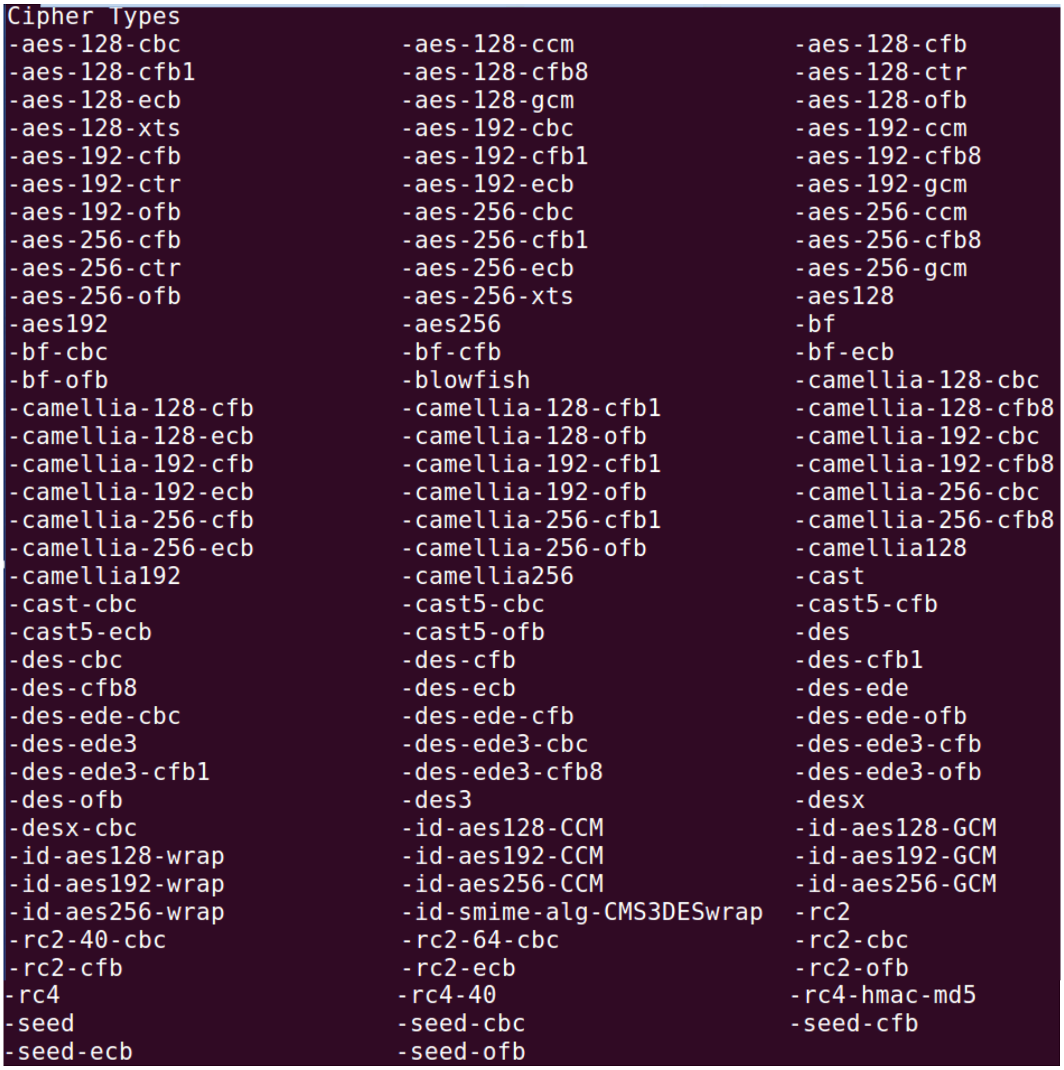
```
[01/31/20]seed@VM:~/.../Lab1$ tr 'abcdefghijklmnopqrstuvwxyz' 'sqwcdyjlxhreobn
 plaintext.txt > ciphertext01.txt
[01/31/20]seed@VM:~/.../Lab1$ head ciphertext01.txt
wnznbsmxzit nifqzdsr: klsft bdvf?
dvpdzft kdxjl ip fld qdtf- sbc knztf-wstd twdbszxnt st fld knzec ldsefl nzjsbx
t s jenqse ldsefl dodzjdbwu.
31 hsbiszu 2020

twxdbfxtft sbc ldsefl siflnzxfxdt sznibc fld knzec szd zswxbj fn lsef fld tpzd
 mxzit flsf dodzjdc xb fld wlxbdtd wxfu ny kilsb xb cdwdoqdz. flnitsbct ny pdn
cu wnbfzswfdc fld bdk wnznbsmxzit, klxwl wsitdt zdtpxzsfnzu xeebdtt. fld cdsfl
3, sbc xt zxtxbj csxeu. nb 30 hsbiszu, fld knzec ldsefl nzjsbxasfxnb (kln) cdw
zdsr s piqexw-ldsefl dodzjdbwu ny xbfdzbsfxnbse wnbwdzb  sb seszo xf zdtdzmdt
f pntd s zxtr fn oiefxped wnibfzxdt sbc klxwl zdgixzdt s wnnzcxbsfdc xbfdzbsfx
wziwxse cdfsxet sqnif fld mxzit sbc lnk xf tpzdsct szd tfxee ibrbnkb, qif dvpd
dzxbj qdtf- sbc knztf-wstd twdbszxnt nb fld qstxt ny pzdmxnit dpxcdoxwt sbc kl
sezdscu rbnk.

lnk osbu pdnped kxee fld mxzit xbydwf?
wlxbdtd siflnzxfxdt lsmd enwrdc cnkb wxfxdt sf fld wdbfzd ny fld dpxcdoxw, sbc
dzd gixwr fn tlszd csfs nb fld mxzit kxfl fld knzec ldsefl nzjsbxasfxnb sbc zd
 fld wstd bioqdzt lsmd qddb zxtxbj, sbc tizjdc pstf 9,000 xb fld pstf csu, ont
flxt lst edc fn nbd pzdcxwfxnb flsf fld mxzit wniec xbydwf sqnif 39,000 ny fld
nped exmxbj xb fld zdjxnb ny kilsb. xf tddot exrd fld mxzit lst jnf nif ny lsb
zdsc fnn ysz, fnn gixwreu fn zdseeu qd wnbfsxbdc, tsut xsb oswrsu, s mxznenjxt
ztxfu ny giddbtesbc xb qzxtqsbd, sitfzsexs.
xb fld qdtf wstd, ydkdz pdnped kxee qd xbydwfdc qdwsitd fld dyydwft ny fld wnb
kxee tfszf rxwrxbj xb, tsut qdb wnkexbj, sb dpxcdoxnenjxtf sf fld ibxmdztxfu n
if xft fnn dszeu fn fdee kldfldz dyynzft fn giszsbfxbd pdnped, sbc fld kxcdtpz
d ostrt, szd knzrxbj. fld xbwiqsfxnb pdzxnc ynz fld mxzit  ip fn 14 csut  xt e
f wnbfzne odstizdt lsmd qddb xb peswd, ld tsut.
[01/31/20]seed@VM:~/.../Lab1$
```



```
[01/31/20]seed@VM:~/.../Lab1$ tr 'fld' 'the' < ciphertext01.txt > out.txt
[01/31/20]seed@VM:~/.../Lab1$ head out.txt
wnznbsmxzit nitqzesr: khstt bevt?
evpeztt kexjh ip the qett- sbc knztt-wste twebszxnt st the knzec heseth nzjsbxa
t s jenqse heseth eoezjebwu.
31 hsbiszu 2020

twxebtxttt sbc heseth sithnzxtxet sznibc the knzec sze zswxbj tn hset the tpzes
 mxzit thst eoezjec xb the whxbete wxtu ny kihsb xb ceweoqez. thnitsbct ny penp
cu wnbtzswtec the bek wnznbsmxzit, khxwh wsitet zetpxzstnzu xeebett. the cesth
3, sbc xt zxtxbj csxeu. nb 30 hsbiszu, the knzec heseth nzjsbxastxnb (khn) cewe
zesr s piqexw-heseth eoezjebwu ny xbtezbstxnbse wnbwezb  sb seszo xt zetezmet y
t pnte s zxtr tn oietxpee wnibtzxet sbc khxwh zegixzet s wnnzcxbstec xbtezbstxn
wziwxse cetsxet sqnit the mxzit sbc hnk xt tpzesct sze ttxee ibrbnkb, qit evpez
ezxbj qett- sbc knztt-wste twebszxnt nb the qstxt ny pzemxnit epxceoxwt sbc khs
sezescu rbnk.

hnk osbu penpee kxee the mxzit xbyewt?
whxbete sithnzxtxet hsme enwrec cnkb wxtxet st the webtze ny the epxceoxw, sbc
eze gixwr tn thsze csts nb the mxzit kxth the knzec heseth nzjsbxastxnb sbc zet
 the wste bioqezt hsme qeeb zxtxbj, sbc tizjec pstt 9,000 xb the pstt csu, ontt
thxt hst eec tn nbe pzecxwtxnb thst the mxzit wniec xbyewt sqnit 39,000 ny the
npee exmxbj xb the zejxnb ny kihsb. xt teeot exre the mxzit hst jnt nit ny hsbc
zesc tnn ysz, tnn gixwreu tn zeseeu qe wnbtsxbec, tsut xsb oswrsu, s mxznenjxtt
ztxtu ny gieebtesbc xb qzxtqsbe, sittzsexs.
xb the qett wste, yekez penpee kxee qe xbyewtec qewsite the eyyewtt ny the wnbt
kxee ttszt rxwrxbj xb, tsut qeb wnkexbj, sb epxceoxnenjxtt st the ibxmeztxtu ny
it xtt tnn eszeu tn teee khethez eyynztt tn giszsbtxbe penpee, sbc the kxcetpze
e ostrt, sze knzrxbj. the xbwiqstxnb pezxnc ynz the mxzit  ip tn 14 csut  xt en
t wnbtzne oestizet hsme qeeb xb peswe, he tsut.
[01/31/20]seed@VM:~/.../Lab1$
```

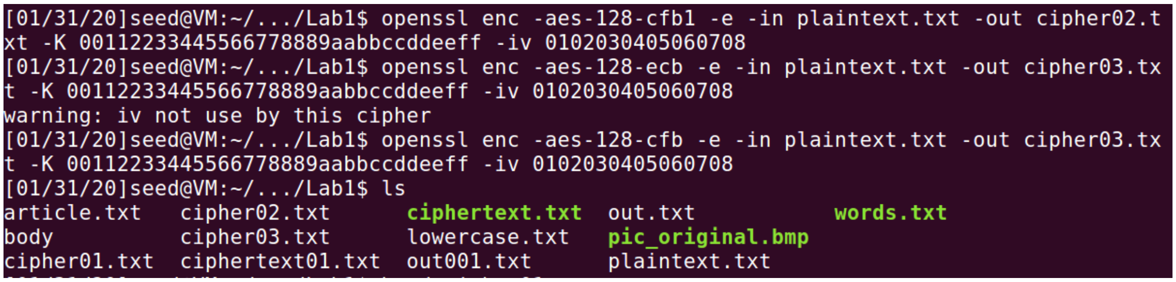
Task 2: Encryption using Different Ciphers and Modes

The following are the available ciphertypes:



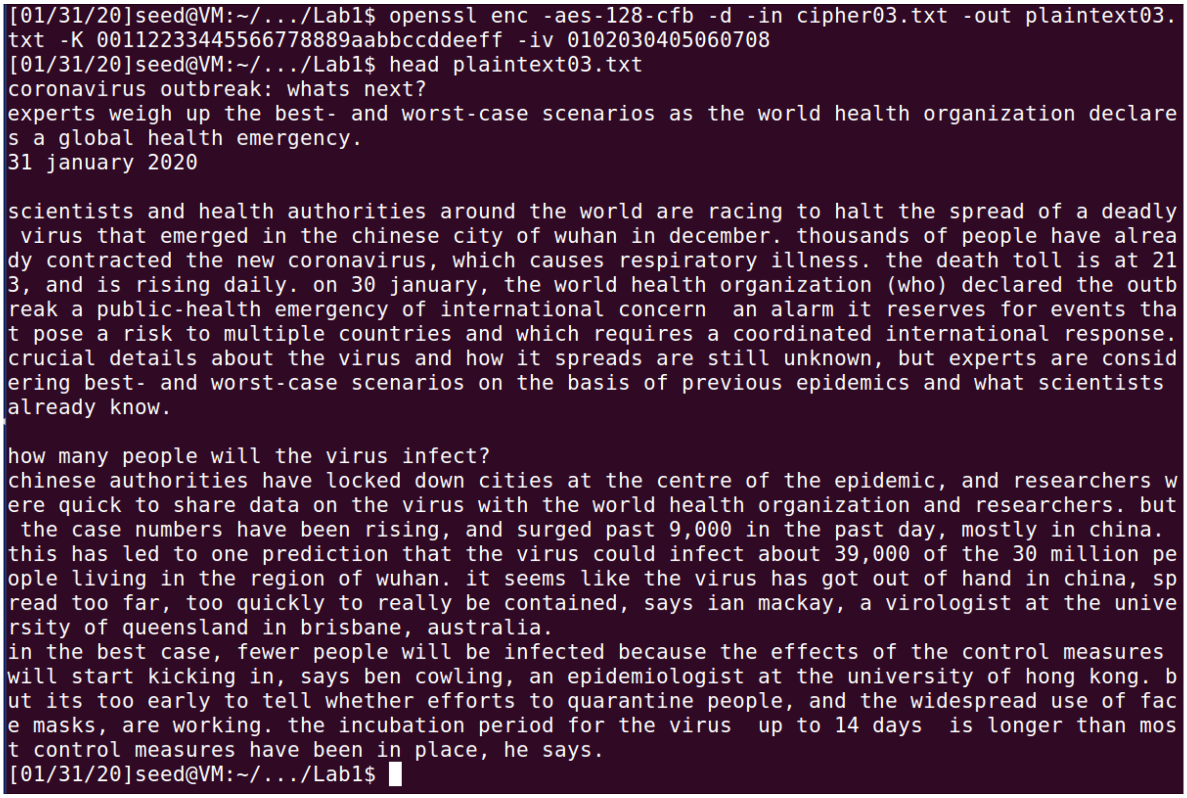
```
Cipher Types
-aes-128-cbc               -aes-128-ccm               -aes-128-cfb
-aes-128-cfb1              -aes-128-cfb8              -aes-128-ctr
-aes-128-ecb               -aes-128-gcm               -aes-128-ofb
-aes-128-xts               -aes-192-cbc               -aes-192-ccm
-aes-192-cfb               -aes-192-cfb1              -aes-192-cfb8
-aes-192-ctr               -aes-192-ecb               -aes-192-gcm
-aes-192-ofb               -aes-256-cbc               -aes-256-ccm
-aes-256-cfb               -aes-256-cfb1              -aes-256-cfb8
-aes-256-ctr               -aes-256-ecb               -aes-256-gcm
-aes-256-ofb               -aes-256-xts               -aes128
-aes192                    -aes256                    -bf
-bf-cbc                    -bf-cfb                    -bf-ecb
-bf-ofb                    -blowfish                  -camellia-128-cbc
-camellia-128-cfb          -camellia-128-cfb1         -camellia-128-cfb8
-camellia-128-ecb          -camellia-128-ofb          -camellia-192-cbc
-camellia-192-cfb          -camellia-192-cfb1         -camellia-192-cfb8
-camellia-192-ecb          -camellia-192-ofb          -camellia-256-cbc
-camellia-256-cfb          -camellia-256-cfb1         -camellia-256-cfb8
-camellia-256-ecb          -camellia-256-ofb          -camellia128
-camellia192               -camellia256               -cast
-cast-cbc                  -cast5-cbc                 -cast5-cfb
-cast5-ecb                 -cast5-ofb                 -des
-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
```

Using openssl for -aes-128-cbc, -aes-128-cfb, -bf-bcb:



```
[01/31/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-cfb1 -e -in plaintext.txt -out cipher02.t
xt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[01/31/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-ecb -e -in plaintext.txt -out cipher03.tx
t -K 00112233445566778889aabbccddeeff -iv 0102030405060708
warning: iv not use by this cipher
[01/31/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-cfb -e -in plaintext.txt -out cipher03.tx
t -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[01/31/20]seed@VM:~/.../Lab1$ ls
article.txt   cipher02.txt      ciphertext.txt  out.txt           words.txt
body          cipher03.txt      lowercase.txt   pic_original.bmp
cipher01.txt  ciphertext01.txt  out001.txt      plaintext.txt
```

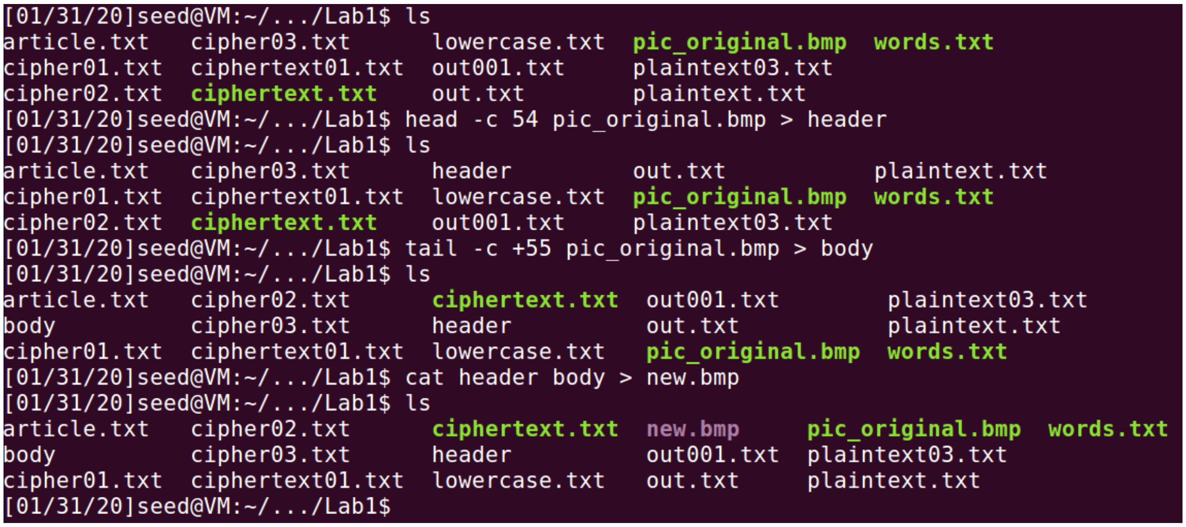
Also trying decryption:



```
[01/31/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-cfb -d -in cipher03.txt -out plaintext03.
txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
[01/31/20]seed@VM:~/.../Lab1$ head plaintext03.txt
coronavirus outbreak: whats next?
experts weigh up the best- and worst-case scenarios as the world health organization declare
s a global health emergency.
31 january 2020

scientists and health authorities around the world are racing to halt the spread of a deadly
 virus that emerged in the chinese city of wuhan in december. thousands of people have alrea
dy contracted the new coronavirus, which causes respiratory illness. the death toll is at 21
3, and is rising daily. on 30 january, the world health organization (who) declared the outb
reak a public-health emergency of international concern  an alarm it reserves for events tha
t pose a risk to multiple countries and which requires a coordinated international response.
crucial details about the virus and how it spreads are still unknown, but experts are consid
ering best- and worst-case scenarios on the basis of previous epidemics and what scientists
already know.

how many people will the virus infect?
chinese authorities have locked down cities at the centre of the epidemic, and researchers w
ere quick to share data on the virus with the world health organization and researchers. but
 the case numbers have been rising, and surged past 9,000 in the past day, mostly in china.
this has led to one prediction that the virus could infect about 39,000 of the 30 million pe
ople living in the region of wuhan. it seems like the virus has got out of hand in china, sp
read too far, too quickly to really be contained, says ian mackay, a virologist at the unive
rsity of queensland in brisbane, australia.
in the best case, fewer people will be infected because the effects of the control measures
will start kicking in, says ben cowling, an epidemiologist at the university of hong kong. b
ut its too early to tell whether efforts to quarantine people, and the widespread use of fac
e masks, are working. the incubation period for the virus  up to 14 days  is longer than mos
t control measures have been in place, he says.
[01/31/20]seed@VM:~/.../Lab1$ 
```

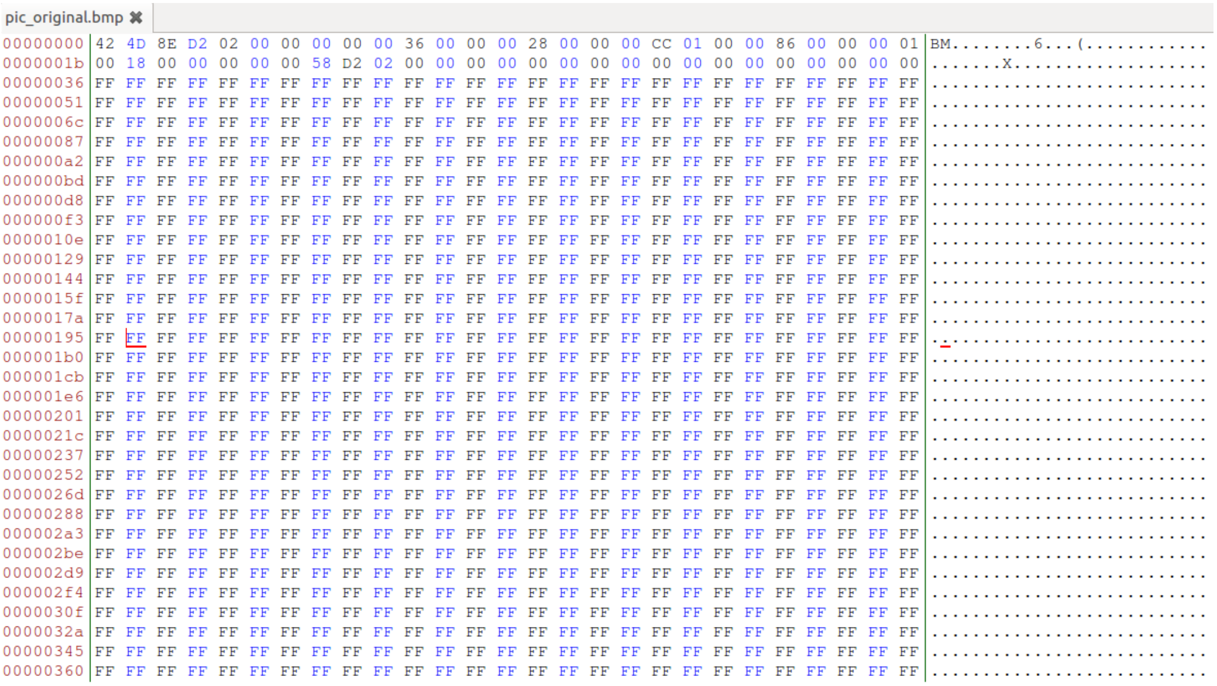
Task 3: Encryption using Different Ciphers and Modes

(Step 1)



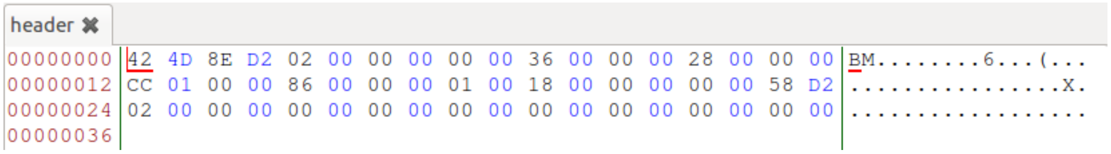
```
[01/31/20]seed@VM:~/.../Lab1$ ls
article.txt   cipher03.txt      lowercase.txt  pic_original.bmp  words.txt
cipher01.txt  ciphertext01.txt  out001.txt     plaintext03.txt
cipher02.txt  ciphertext.txt    out.txt        plaintext.txt
[01/31/20]seed@VM:~/.../Lab1$ head -c 54 pic_original.bmp > header
[01/31/20]seed@VM:~/.../Lab1$ ls
article.txt   cipher03.txt      header         out.txt           plaintext.txt
cipher01.txt  ciphertext01.txt  lowercase.txt  pic_original.bmp  words.txt
cipher02.txt  ciphertext.txt    out001.txt     plaintext03.txt
[01/31/20]seed@VM:~/.../Lab1$ tail -c +55 pic_original.bmp > body
[01/31/20]seed@VM:~/.../Lab1$ ls
article.txt   cipher02.txt      ciphertext.txt  out001.txt        plaintext03.txt
body          cipher03.txt      header          out.txt           plaintext.txt
cipher01.txt  ciphertext01.txt  lowercase.txt   pic_original.bmp  words.txt
[01/31/20]seed@VM:~/.../Lab1$ cat header body > new.bmp
[01/31/20]seed@VM:~/.../Lab1$ ls
article.txt   cipher02.txt      ciphertext.txt  new.bmp     pic_original.bmp  words.txt
body          cipher03.txt      header          out001.txt  plaintext03.txt
cipher01.txt  ciphertext01.txt  lowercase.txt   out.txt     plaintext.txt
[01/31/20]seed@VM:~/.../Lab1$
```

Original Picture:



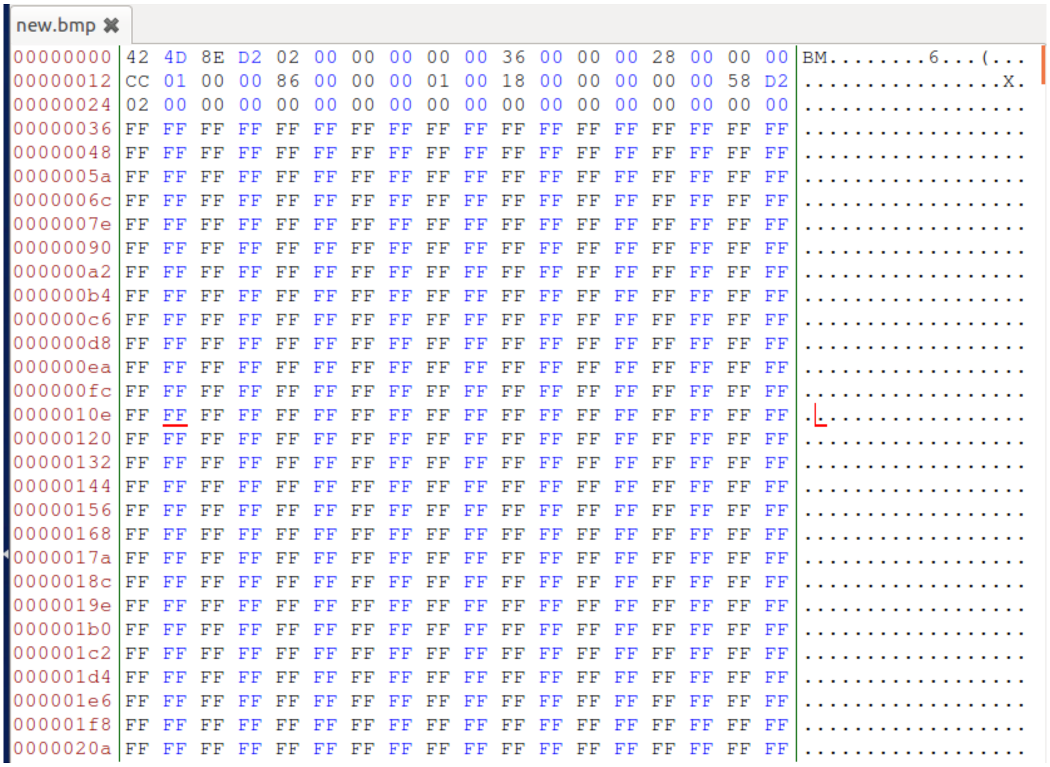
```
pic_original.bmp ✖
00000000 42 4D 8E D2 02 00 00 00 00 00 36 00 00 00 28 00 00 00 CC 01 00 00 86 00 00 00 01 BM........6...(............
0000001b 00 18 00 00 00 00 00 58 D2 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .......X...................
00000036 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000051 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
0000006c FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000087 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000000a2 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000000bd FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000000d8 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000000f3 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
0000010e FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000129 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000144 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
0000015f FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
0000017a FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000195 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000001b0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000001cb FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000001e6 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000201 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
0000021c FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000237 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000252 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
0000026d FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000288 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000002a3 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000002be FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000002d9 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
000002f4 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
0000030f FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
0000032a FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000345 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
00000360 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ...........................
```

Header without the padding:



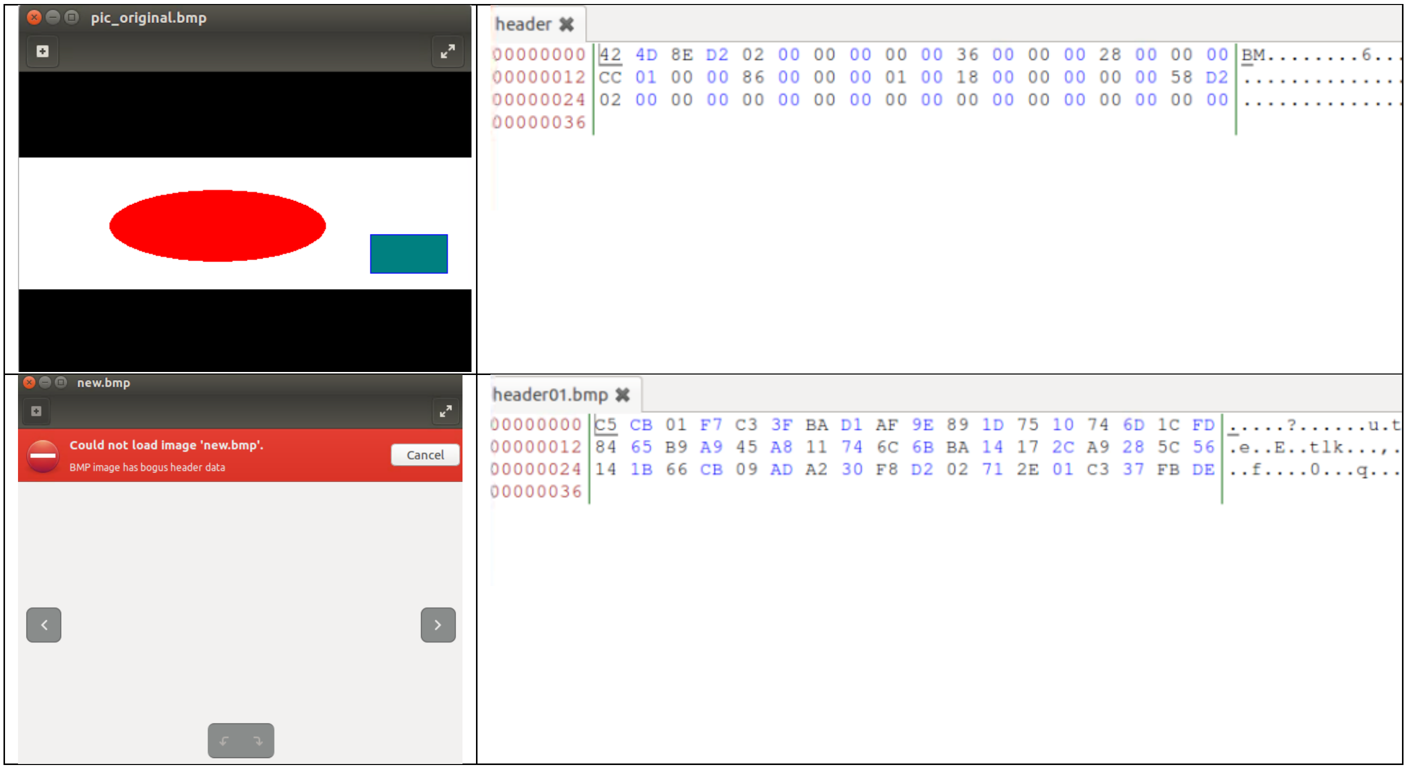
```
header ✖
00000000 42 4D 8E D2 02 00 00 00 00 00 36 00 00 00 28 00 00 00 BM........6...(...
00000012 CC 01 00 00 86 00 00 00 01 00 18 00 00 00 00 00 58 D2 ................X.
00000024 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..................
00000036
```

Encrypted Picture:



```
new.bmp ✖
00000000 42 4D 8E D2 02 00 00 00 00 00 36 00 00 00 28 00 00 00 BM........6...(...
00000012 CC 01 00 00 86 00 00 00 01 00 18 00 00 00 00 00 58 D2 ................X.
00000024 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..................
00000036 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
00000048 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
0000005a FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
0000006c FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
0000007e FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
00000090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000000a2 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000000b4 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000000c6 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000000d8 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000000ea FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000000fc FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
0000010e FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
00000120 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
00000132 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
00000144 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
00000156 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
00000168 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
0000017a FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
0000018c FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
0000019e FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000001b0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000001c2 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000001d4 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000001e6 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
000001f8 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
0000020a FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ..................
```

(Step 2) Below are the encrypted and decrypted



```
00000000 42 4D 8E D2 02 00 00 00 00 00 36 00 00 00 28 00 00 00 BM........6...
00000012 CC 01 00 00 86 00 00 00 01 00 18 00 00 00 00 00 58 D2 ..............
00000024 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..............
00000036
```

Could not load image 'new.bmp'.

BMP image has bogus header data

```
00000000 C5 CB 01 F7 C3 3F BA D1 AF 9E 89 1D 75 10 74 6D 1C FD .....?......u.t
00000012 84 65 B9 A9 45 A8 11 74 6C 6B BA 14 17 2C A9 28 5C 56 .e..E..tlk...,.
00000024 14 1B 66 CB 09 AD A2 30 F8 D2 02 71 2E 01 C3 37 FB DE ..f....0...q...
00000036
```

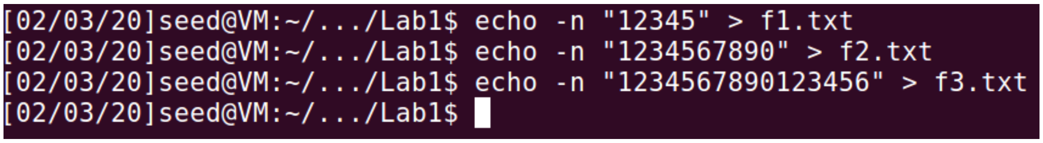
Task 4: Padding

(Step 1) CBC, OFB and CFB are similar, they all have padding; ECB has padding. However, OFB and CFB only requires encryption and not decryption.

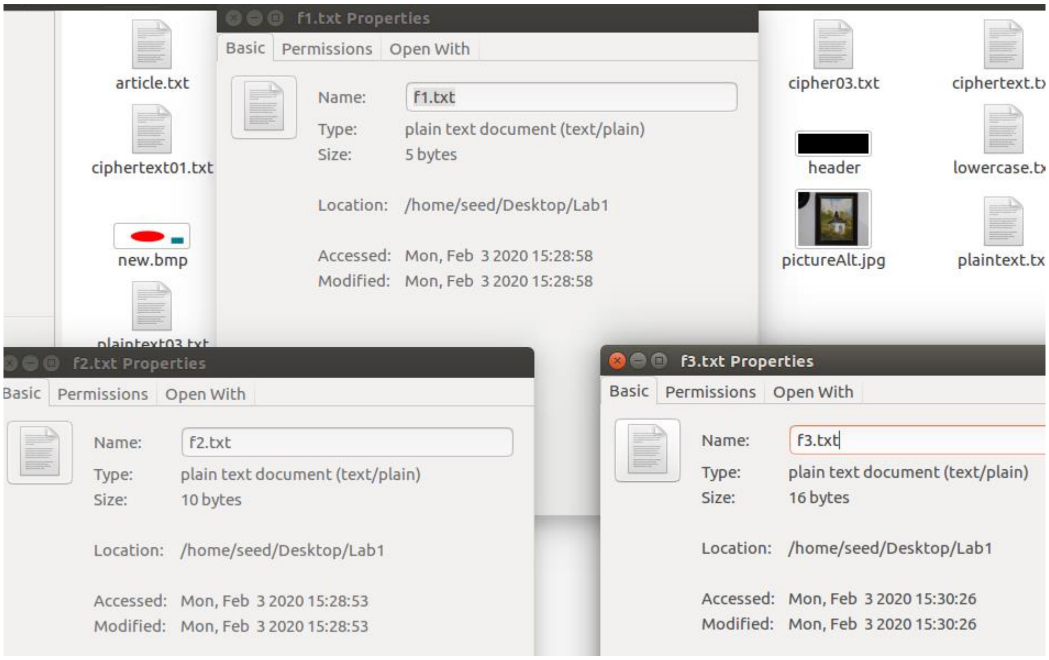
(Step 2) Commands used for encryption and decryption:

```
openssl enc -aes-128-cbc -e -in f1.txt -out f1.enc.txt
openssl enc -aes-128-cbc -d -nopad -in f1.enc.txt -out f1.plain.txt
```

Three files of 5byes, 10 bytes and 16 bytes are created as follows:



```
[02/03/20]seed@VM:~/.../Lab1$ echo -n "12345" > f1.txt
[02/03/20]seed@VM:~/.../Lab1$ echo -n "1234567890" > f2.txt
[02/03/20]seed@VM:~/.../Lab1$ echo -n "1234567890123456" > f3.txt
[02/03/20]seed@VM:~/.../Lab1$ 
```



f1.txt Properties

Basic | Permissions | Open With

Name: f1.txt
Type: plain text document (text/plain)
Size: 5 bytes

Location: /home/seed/Desktop/Lab1

Accessed: Mon, Feb 3 2020 15:28:58
Modified: Mon, Feb 3 2020 15:28:58

f2.txt Properties

Basic | Permissions | Open With

Name: f2.txt
Type: plain text document (text/plain)
Size: 10 bytes

Location: /home/seed/Desktop/Lab1

Accessed: Mon, Feb 3 2020 15:28:53
Modified: Mon, Feb 3 2020 15:28:53

f3.txt Properties

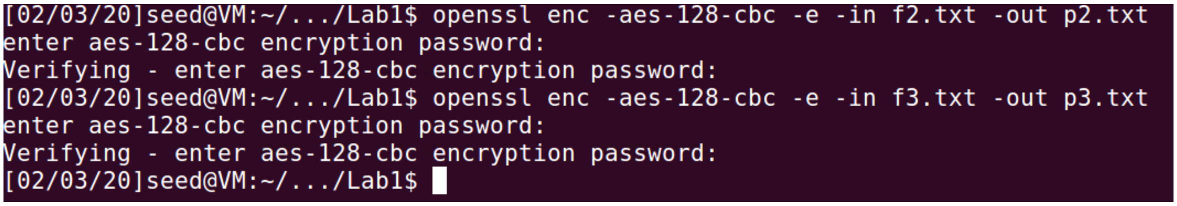
Basic | Permissions | Open With

Name: f3.txt
Type: plain text document (text/plain)
Size: 16 bytes

Location: /home/seed/Desktop/Lab1

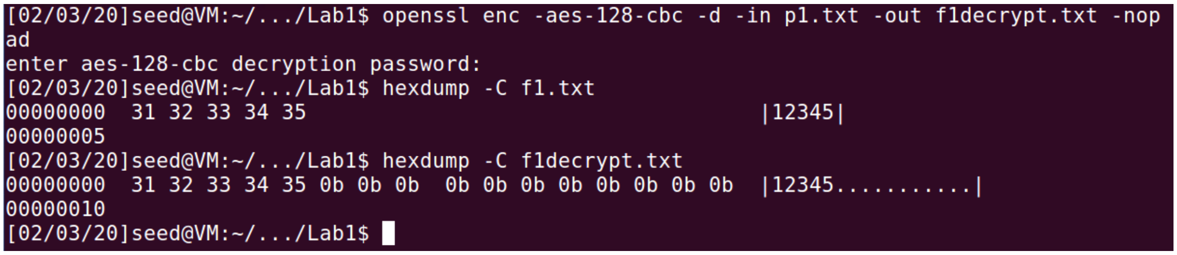
Accessed: Mon, Feb 3 2020 15:30:26
Modified: Mon, Feb 3 2020 15:30:26

Using `openssl enc -aes-128-cbc -e` to encrypt:

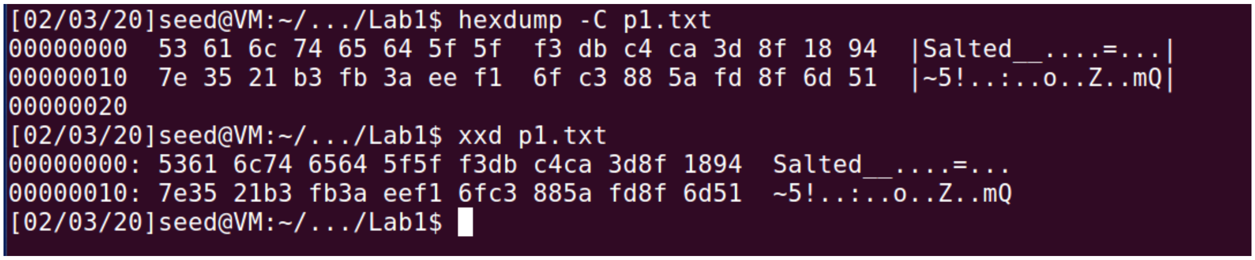


```
[02/03/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-cbc -e -in f2.txt -out p2.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[02/03/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-cbc -e -in f3.txt -out p3.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
[02/03/20]seed@VM:~/.../Lab1$
```

Using `-nopad`:



```
[02/03/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-cbc -d -in p1.txt -out f1decrypt.txt -nop
ad
enter aes-128-cbc decryption password:
[02/03/20]seed@VM:~/.../Lab1$ hexdump -C f1.txt
00000000  31 32 33 34 35                                    |12345|
00000005
[02/03/20]seed@VM:~/.../Lab1$ hexdump -C f1decrypt.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b 0b  |12345...........|
00000010
[02/03/20]seed@VM:~/.../Lab1$
```



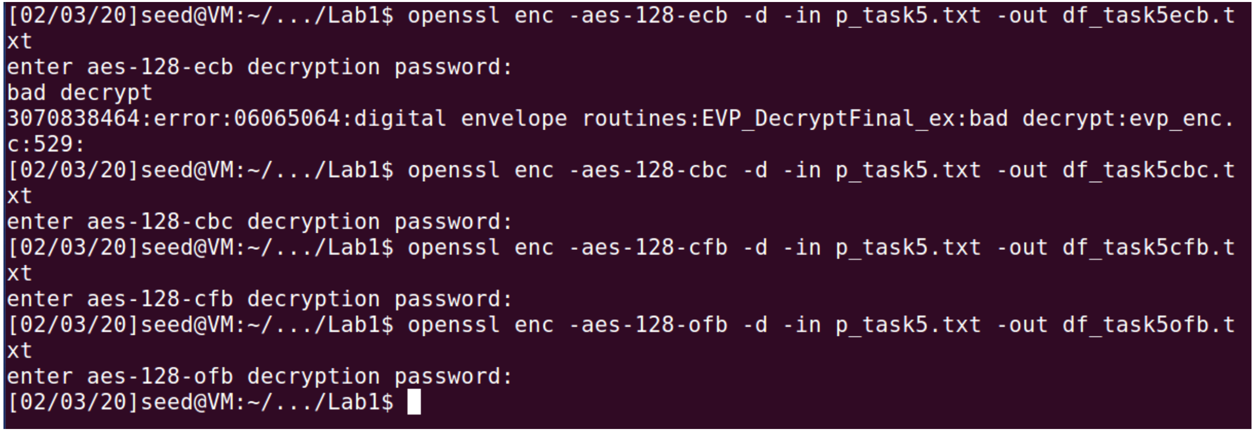
```
[02/03/20]seed@VM:~/.../Lab1$ hexdump -C p1.txt
00000000  53 61 6c 74 65 64 5f 5f  f3 db c4 ca 3d 8f 18 94  |Salted__....=...|
00000010  7e 35 21 b3 fb 3a ee f1  6f c3 88 5a fd 8f 6d 51  |~5!..:..o..Z..mQ|
00000020
[02/03/20]seed@VM:~/.../Lab1$ xxd p1.txt
00000000: 5361 6c74 6564 5f5f f3db c4ca 3d8f 1894  Salted__....=...
00000010: 7e35 21b3 fb3a eef1 6fc3 885a fd8f 6d51  ~5!..:..o..Z..mQ
[02/03/20]seed@VM:~/.../Lab1$
```

Task 5: Encryption mask – Corrupted Ciphertext

ECB	In ECB, during encryption identical blocks are encrypted into cipher blocks. But means that the repeated message is easily recognized. The order of the encrypted blocks can be changed. This is typically used to encrypt data in a single block. When corrupt, it is abandoned and hence shouldn't be used.
CBC	In CBC initialization vector (IV) is used in encryption. But identical blocks will produces different results. So, for a corrupted file, all blocks will be affected.

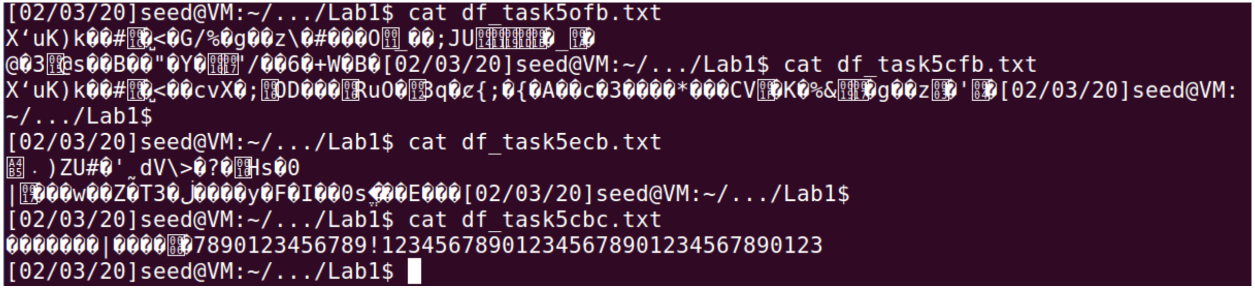
OFB	OFB allows stream encryption. What you do for encryption and decryption are very similar. An error in a block does not affect the other blocks. However, when corrupted in a block, it would not affect other blocks.
CFB	CFB can perform stream encryption. What you do for encryption and decryption are very similar. When corrupted in a block, it affects other blocks as well.

Creating the 64 bytes long text file, encrypting file using the AES-128 cipher. I corrupted the 30th byte in the encrypted file using the bless hex editor, and decrypted the corrupted file using key and IV.



```
[02/03/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-ecb -d -in p_task5.txt -out df_task5ecb.t
xt
enter aes-128-ecb decryption password:
bad decrypt
3070838464:error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt:evp_enc.
c:529:
[02/03/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-cbc -d -in p_task5.txt -out df_task5cbc.t
xt
enter aes-128-cbc decryption password:
[02/03/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-cfb -d -in p_task5.txt -out df_task5cfb.t
xt
enter aes-128-cfb decryption password:
[02/03/20]seed@VM:~/.../Lab1$ openssl enc -aes-128-ofb -d -in p_task5.txt -out df_task5ofb.t
xt
enter aes-128-ofb decryption password:
[02/03/20]seed@VM:~/.../Lab1$ 
```

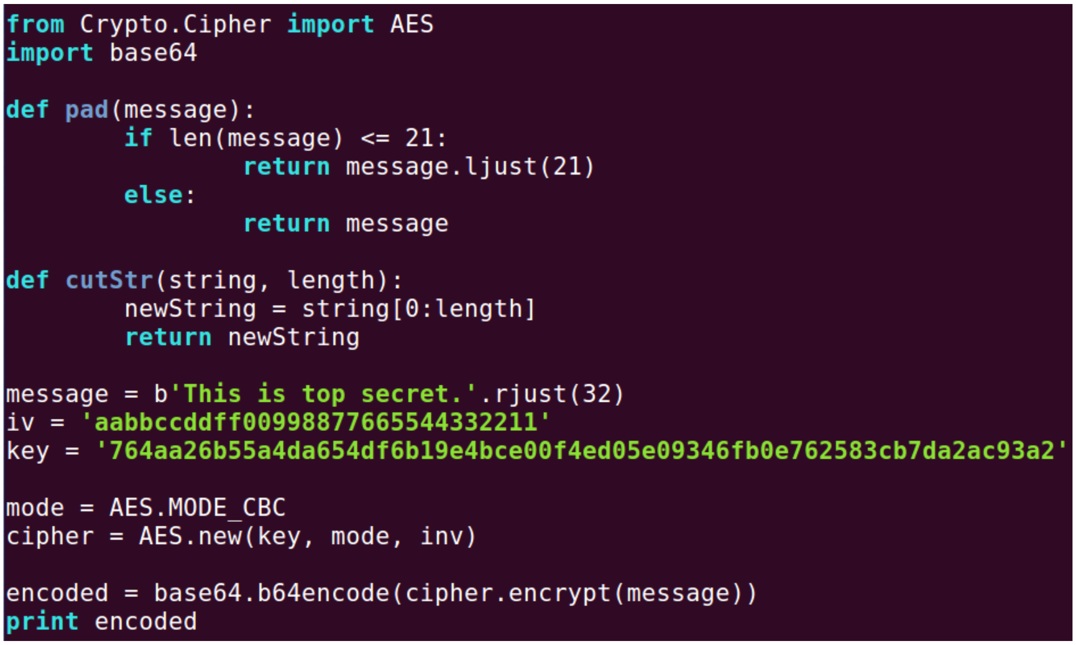
Decryption of corrupted file using ECB(task5ecb), CBC(task5cbc), CFB(task5cfb), and OFB (task5ofb):



```
[02/03/20]seed@VM:~/.../Lab1$ cat df_task5ofb.txt
X'uK)k��#��<�G/%�g��z\�#���O� ��;JU����_��
@�3�s��B��"�Y��'/��6�+W�B�[02/03/20]seed@VM:~/.../Lab1$ cat df_task5cfb.txt
X'uK)k��#��<��cvX�;�DD����Ru0��3q�c{;�{�A��c�3����*���CV��K�%&��g��z��'��[02/03/20]seed@VM:
~/.../Lab1$
[02/03/20]seed@VM:~/.../Lab1$ cat df_task5ecb.txt
�.)ZU#�' dV\>�?��Hs�0
|����w��Z�T3�J����y�F�I��0s���E���[02/03/20]seed@VM:~/.../Lab1$
[02/03/20]seed@VM:~/.../Lab1$ cat df_task5cbc.txt
�������|�����7890123456789!1234567890123456789012345678901234567890123
[02/03/20]seed@VM:~/.../Lab1$ 
```

As seen above, the best decrypted results come from CBC (task5cbc).

Task 7: Programming using the Crypto Library



```
from Crypto.Cipher import AES
import base64

def pad(message):
        if len(message) <= 21:
                return message.ljust(21)
        else:
                return message

def cutStr(string, length):
        newString = string[0:length]
        return newString

message = b'This is top secret.'.rjust(32)
iv = 'aabbccddff00998877665544332211'
key = '764aa26b55a4da654df6b19e4bce00f4ed05e09346fb0e762583cb7da2ac93a2'

mode = AES.MODE_CBC
cipher = AES.new(key, mode, inv)

encoded = base64.b64encode(cipher.encrypt(message))
print encoded
```