

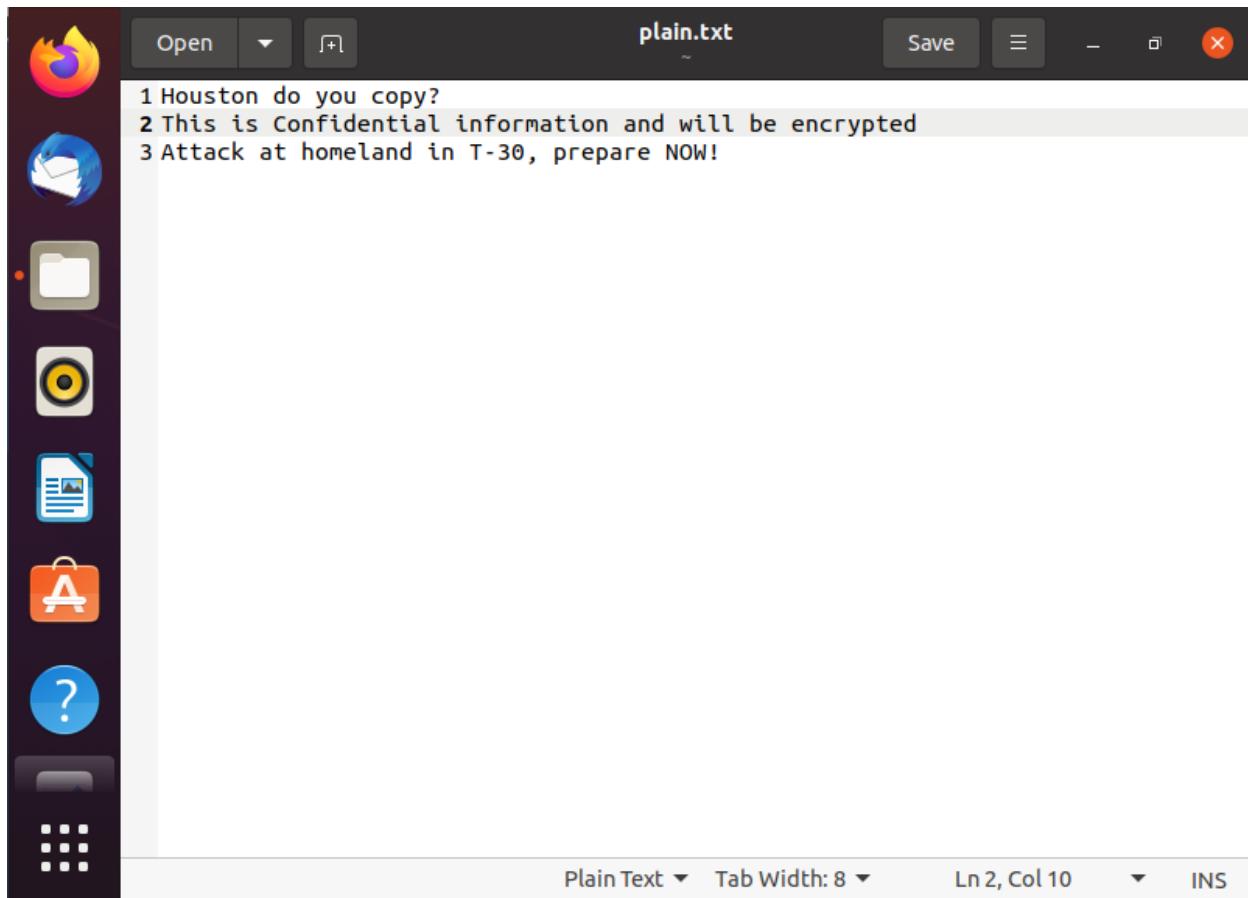
Aditya Motwani
2019130040
TE Comps Batch B

CSS Experiment 3

Aim: To get familiar with the concepts in secret-key encryption. To gain first-hand experience on encryption algorithms, encryption modes, paddings, and initial vector (IV) by using tools and writing programs to encrypt/decrypt messages.

Task 1:

Plain Text:



The screenshot shows a terminal window titled "plain.txt". The window has a dark theme with light-colored text. The text content is as follows:

```
1 Houston do you copy?  
2 This is Confidential information and will be encrypted  
3 Attack at homeland in T-30, prepare NOW!
```

The terminal interface includes standard file operations like "Open" and "Save", and status indicators for "Plain Text", "Tab Width: 8", "Ln 2, Col 10", and "INS". A vertical toolbar on the left contains icons for file operations such as Open, Save, Copy, Paste, Find, and Delete.

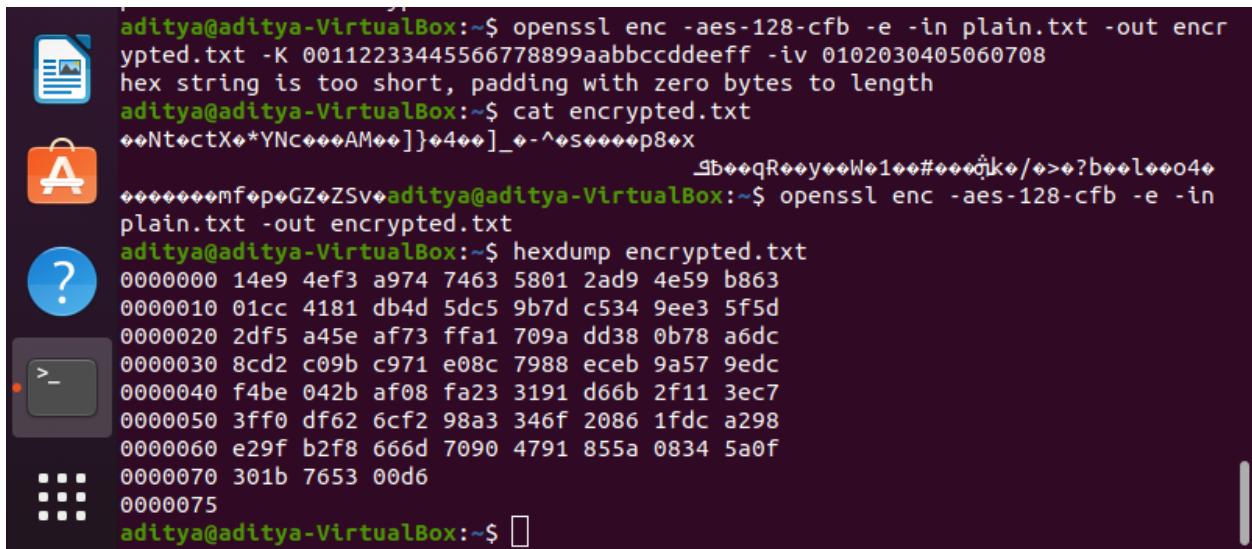
1) Encryption using aes-128-ecb

```
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ecb -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~$ cat encrypted.txt
♦'♦)7♦g♦"♦@♦"♦♦♦♦禹♦\&4♦♦♦♦♦♦{♦♦+♦♦♦{♦♦♦♦
♦♦7♦U♦!♦$♦en^♦Z♦ver♦/♦♦♦♦
o♦X*♦♦Z♦I♦N♦G♦l♦I
♦Y♦aditya@aditya-VirtualBox:~$ hexdump encrypted.txt
0000000 27f6 2988 8437 671e 18d5 009a 22f4 8440
0000010 ba22 f5c3 8002 e70e baa6 d417 265c f534
0000020 02b0 09c2 848e 9292 7b6f ae83 062b b1b9
0000030 64a7 a47b f9ee 0b17 810f e804 37b2 bfda
0000040 b1c6 ae1d fb21 1b24 ca85 f79a 6e86 135e
0000050 bf8c 9a5a 080f 768b 72e6 b9e5 c5ee 2f95
0000060 9492 cadd a2e6 1c6f 58cf 932a c5bf e2bb
0000070 1e49 4ed5 c6dd be93 d56c 0b49 59e8 ffa0
0000080
```

2) Encryption using aes-128-cbc

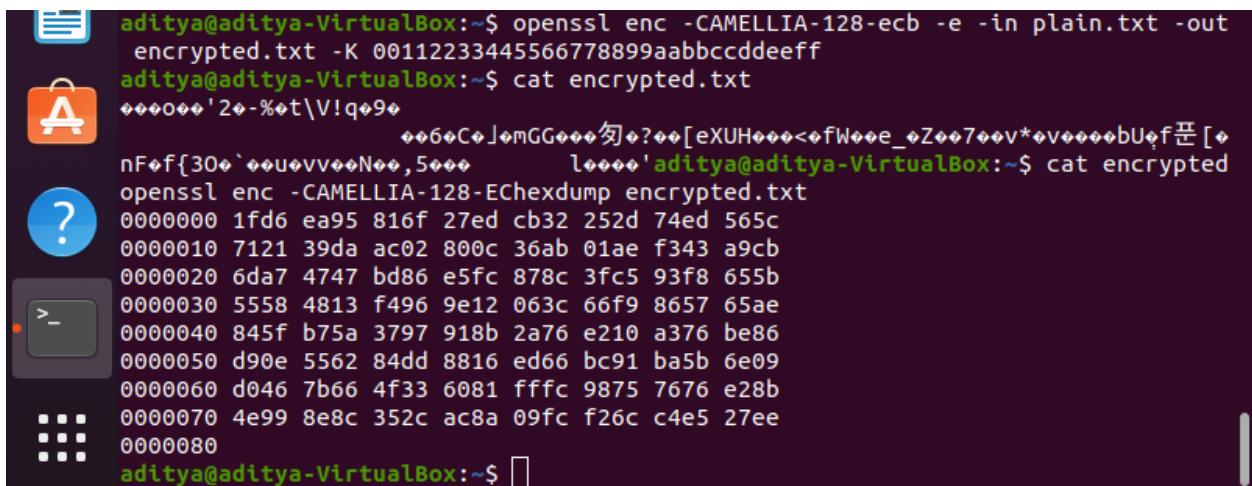
```
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cbc -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat encrypted.txt
PqmX♦♦♦♦=♦♦bY]?♦♦♦♦i♦♦=F♦h♦*♦;0♦2♦e♦♦♦[♦<♦8♦
♦♦♦>hH♦B♦♦♦Vx♦R♦♦♦0♦P?l♦ql♦@0/♦♦♦X♦-♦♦♦!♦|♦$♦cMM-G♦Z♦1g:U-7Q;♦aditya@adit
ya-VirtualBox:~$ hexdump encrypted.txt
hexdump: encrypted.t: No such file or directory
aditya@aditya-VirtualBox:~$ hexdump encrypted.txt
0000000 7150 586d 93f4 c2a4 873d 62d9 4a59 a93f
0000010 169c 0292 c7b5 8a90 3dd7 fe46 a7c4 2a86
0000020 3bbf 9830 bb32 e11c b365 bde9 06e4 5bd4
0000030 3c9b 38ba 0a81 eac3 3ed7 1168 fa48 8d42
0000040 e28b 7856 8bc0 c152 d7b6 d430 3f50 bd6c
0000050 c471 8bbb 3040 982f dec9 b978 aa03 8f2d
0000060 849f 219b 7c9b fc1e 2493 63a8 4d4d 472d
0000070 d0e0 d931 3a88 2d55 5137 9e3b
d Show Applications
aditya@aditya-VirtualBox:~$
```

3) Encryption using aes-128-cfb



```
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cfb -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat encrypted.txt
♦♦NtctX♦*YNC♦♦AM♦♦]♦4♦]_♦-^♦s♦♦♦♦p8♦x
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cfb -e -in plain.txt -out encrypted.txt
aditya@aditya-VirtualBox:~$ hexdump encrypted.txt
00000000 14e9 4ef3 a974 7463 5801 2ad9 4e59 b863
00000010 01cc 4181 db4d 5dc5 9b7d c534 9ee3 5f5d
00000020 2df5 a45e af73 ffa1 709a dd38 0b78 a6dc
00000030 8cd2 c09b c971 e08c 7988 eceb 9a57 9edc
00000040 f4be 042b af08 fa23 3191 d66b 2f11 3ec7
00000050 3ff0 df62 6cf2 98a3 346f 2086 1fdc a298
00000060 e29f b2f8 666d 7090 4791 855a 0834 5a0f
00000070 301b 7653 00d6
00000075
aditya@aditya-VirtualBox:~$
```

4) Encryption using CAMELLIA-128-ecb



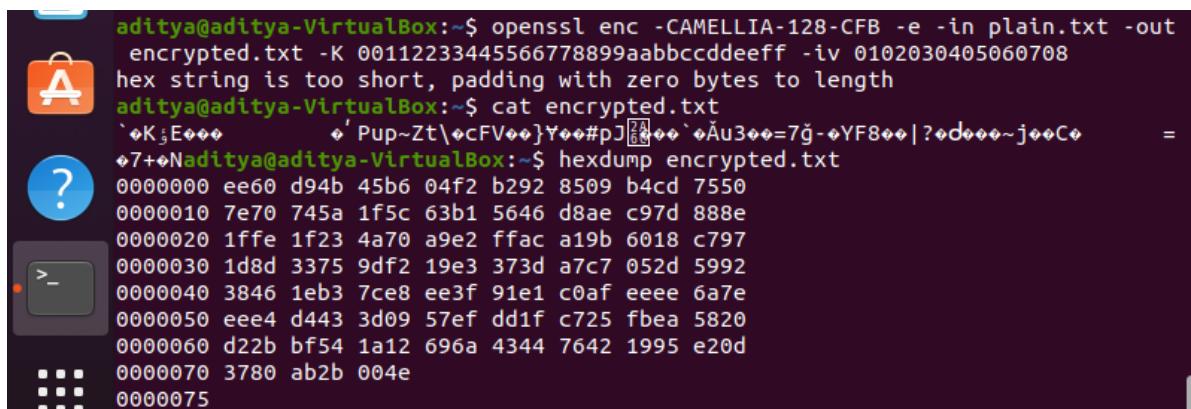
```
aditya@aditya-VirtualBox:~$ openssl enc -CAMELLIA-128-ecb -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~$ cat encrypted.txt
♦♦♦♦'2♦-%♦t\Viq♦9♦
aditya@aditya-VirtualBox:~$ openssl enc -CAMELLIA-128-ECBhexdump encrypted.txt
00000000 1fd6 ea95 816f 27ed cb32 252d 74ed 565c
00000010 7121 39da ac02 800c 36ab 01ae f343 a9cb
00000020 6da7 4747 bd86 e5fc 878c 3fc5 93f8 655b
00000030 5558 4813 f496 9e12 063c 66f9 8657 65ae
00000040 845f b75a 3797 918b 2a76 e210 a376 be86
00000050 d90e 5562 84dd 8816 ed66 bc91 ba5b 6e09
00000060 d046 7b66 4f33 6081 fffc 9875 7676 e28b
00000070 4e99 8e8c 352c ac8a 09fc f26c c4e5 27ee
00000080
aditya@aditya-VirtualBox:~$
```

5) Encryption using CAMELLIA-128-cbc



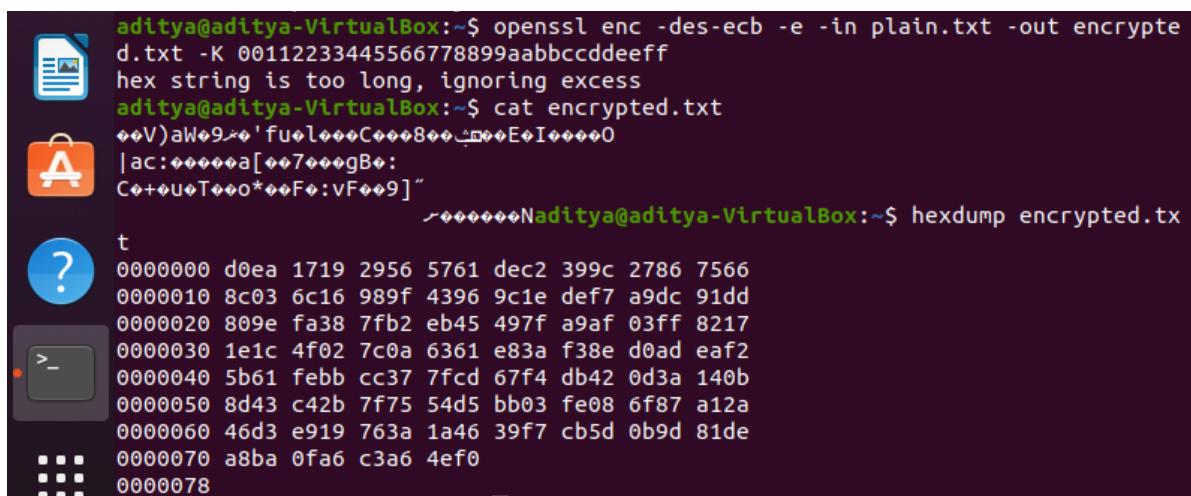
```
aditya@aditya-VirtualBox:~$ openssl enc -CAMELLIA-128-CBC -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat encrypted.txt
Z%o@Poo3oX
    .Voo.8.oooooDXo      .Q>^_oooB$o3oDooo_0ooo,hkUooo").AbooJooo#/Xo@HwyXq3o.Moo7/ooooo6[_oo      #K}oooaditya@aditya-VirtualBox:~$ hexdump encrypted.txt
00000000 255a 40ad ad50 0797 b633 5803 c80c 6f56
00000010 1df6 fc2c 2e38 b11a 95bb a78a 5844 ac17
00000020 f609 3e51 5e81 d85f 04d6 42b0 fe24 93ca
00000030 448e 87a2 5fb8 04fc 3016 13e0 aea5 682c
00000040 556b c497 22fe 2e29 6241 f6d5 1d4a 98a0
00000050 1c03 12c8 2319 1d2f 5810 4098 7748 5879
00000060 3371 d9c7 4d81 d61a 9b11 2f37 cee5 c6fa
00000070 36f0 5f5b b0b8 2309 7d4b e19f 86df cccf
00000080
```

6) Encryption using CAMELLIA-128-cfb



```
aditya@aditya-VirtualBox:~$ openssl enc -CAMELLIA-128-CFB -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat encrypted.txt
`oK;Eooo      .Pup~Zt\oCFVoo}Yooo#pJ[.ooo`oAu3ooo=7g-ooYF8ooo|?odooo~joooCoo
o7+oNaditya@aditya-VirtualBox:~$ hexdump encrypted.txt
00000000 ee60 d94b 45b6 04f2 b292 8509 b4cd 7550
00000010 7e70 745a 1f5c 63b1 5646 d8ae c97d 888e
00000020 1ffe 1f23 4a70 a9e2 ffac a19b 6018 c797
00000030 1d8d 3375 9df2 19e3 373d a7c7 052d 5992
00000040 3846 1eb3 7ce8 ee3f 91e1 c0af eeee 6a7e
00000050 eee4 d443 3d09 57ef dd1f c725 fbea 5820
00000060 d22b bf54 1a12 696a 4344 7642 1995 e20d
00000070 3780 ab2b 004e
00000075
```

7) Encryption using des-128-ecb



```
aditya@aditya-VirtualBox:~$ openssl enc -des-ecb -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff
hex string is too long, ignoring excess
aditya@aditya-VirtualBox:~$ cat encrypted.txt
ooV)aWo9~o'fuoloooCooo8oo.oooEoIooooO
|ac:oooooa[oo7oooBgoo
Co+oUoTooo*ooFo:vFo*9]"o
aditya@aditya-VirtualBox:~$ hexdump encrypted.txt
00000000 d0ea 1719 2956 5761 dec2 399c 2786 7566
00000010 8c03 6c16 989f 4396 9c1e def7 a9dc 91dd
00000020 809e fa38 7fb2 eb45 497f a9af 03ff 8217
00000030 1e1c 4f02 7c0a 6361 e83a f38e d0ad eaf2
00000040 5b61 febb cc37 7fcd 67f4 db42 0d3a 140b
00000050 8d43 c42b 7f75 54d5 bb03 fe08 6f87 a12a
00000060 46d3 e919 763a 1a46 39f7 cb5d 0b9d 81de
00000070 a8ba 0fa6 c3a6 4ef0
00000078
```

8) Encryption using des-128-cbc



```
aditya@aditya-VirtualBox:~$ openssl enc -des-cbc -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too long, ignoring excess
aditya@aditya-VirtualBox:~$ hexdump encrypted.txt
00000000 75aa 6cbb dbf4 5fa6 a222 bc8e 8fbb ff7c
00000010 4ade 2fce dd01 53e1 b196 db64 5dc9 2012
00000020 b610 84ae 0e64 fca1 b48f a719 59da b1ee
00000030 089a 55fe 8f88 80da f50d 0528 6c6d 59ed
00000040 b4ff 12d7 11b8 63bf f137 852a 01f4 c240
00000050 bf87 3643 70f0 4a9e 0223 45f2 df01 7eee
00000060 4ac5 de9f e970 9cde a610 0f04 e424 5e67
00000070 752e 2960 5c51 5110
00000078
aditya@aditya-VirtualBox:~$ cat encrypted.txt
♦(ml♦Y♦♦♦♦c7♦*♦@♦C6♦p♦J#♦E♦~♦J♦p♦~♦$♦g^..u` )Q\Qaditya@aditya-VirtualBox:~$
```

9) Encryption using des-128-cbc



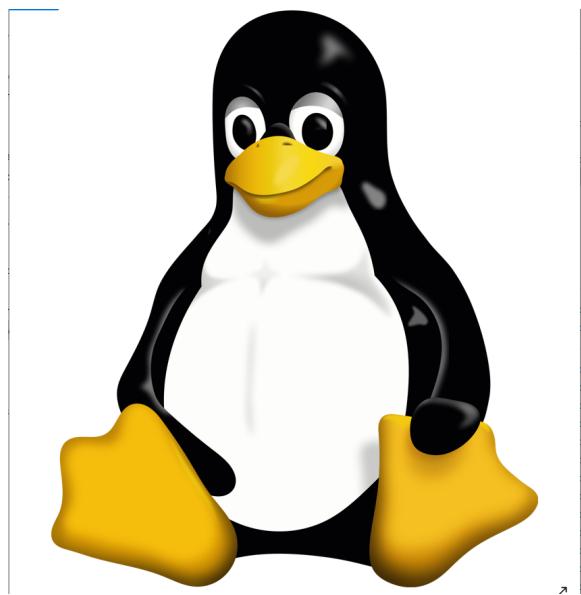
```
aditya@aditya-VirtualBox:~$ openssl enc -des-cfb -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too long, ignoring excess
aditya@aditya-VirtualBox:~$ hexdump encrypted.txt
00000000 6bac ac86 cb6c 3b3d 6d7d 10e1 6a49 c38b
00000010 5a3e 3994 ae39 63b3 91ef 8d7a c313 1f46
00000020 d0c4 d057 d903 75ca 1f1e adbc 0736 b6df
00000030 b37a e11c fa47 2aae 7d33 832c c4df 5c22
00000040 210b af66 3bfa 9af4 88a6 99dd 7278 ce99
00000050 7ec3 22a9 25eb a2c1 a631 cb8c a77d d9bb
00000060 3d8e a3ca d185 8886 3e64 2bbc 34f7 7a89
00000070 ec8a 2c88 00a8
00000075
aditya@aditya-VirtualBox:~$ cat encrypted.txt
♦k♦o♦l♦=; }m♦I♦j♦>Z♦99♦c♦z♦F♦W♦u♦6[♦]♦G♦*3} ,♦" " \ ! f♦;♦♦♦♦X♦Γ♦~♦" ♦%♦1♦♦♦♦
♦}♦=d♦z♦d♦>♦+♦4♦z♦,♦aditya@aditya-VirtualBox:~$
```

Task 2:

```
aditya@aditya-VirtualBox:~$ head -c 54 linux.bmp > ecb.bmp
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ecb -e -in linux.bmp -out output.bin -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~$ tail -c 6868884 output.bin >> ecb.bmp
```

```
aditya@aditya-VirtualBox:~$ head -c 54 linux.bmp > cbc.bmp
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cbc -e -in linux.bmp -out output.bin -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ tail -c 6868884 output.bin >> cbc.bmp
```

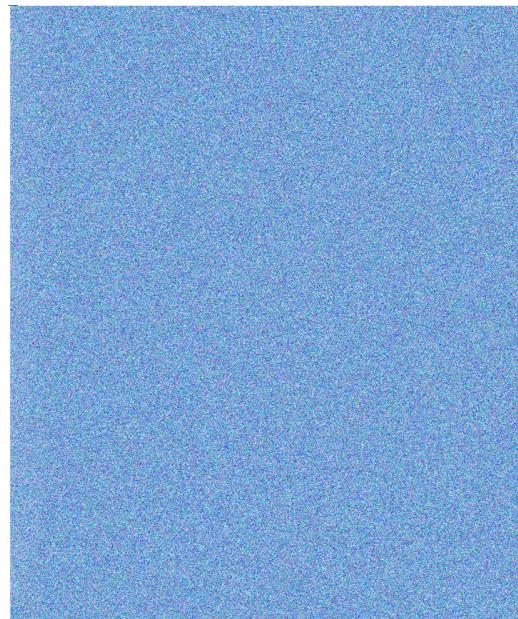
1) Original Image: linux.bmp



2) ECB Encrypted Image: ecb.bmp



3) CBC Encrypted Image: cbc.bmp



Task 3:

1) ECB

```
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ecb -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~$ ghex
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ecb -d -in encrypted.txt -out decrypted.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~$ cat decrypted.txt
Houston do you c•T•fH•r•VU•t••l•fidential information and will be e
ncrypted
Attack at homeland in T-30, prepare NOW!
```

2) CBC

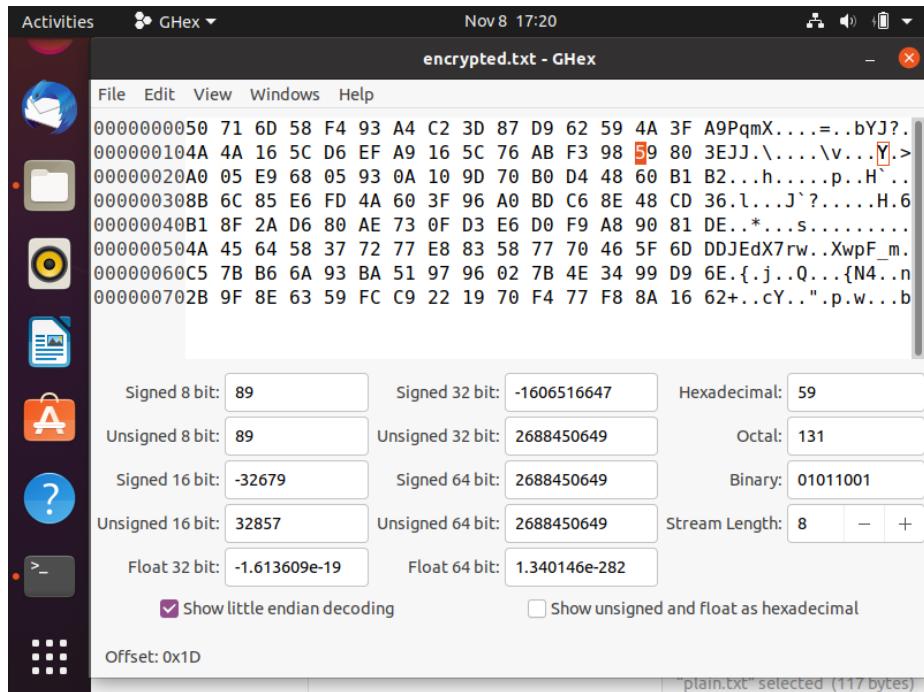
```
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cbc -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ ghex
aditya@aditya-VirtualBox:~$ cat encrypted.txt
PqmX++++=++bYJ?+JJ\++\V++++>++h+
++p++H`++l++J`?++EHa6++*]ks++++++JEdX7rw+XwpF_m++{+j++Q++{N4++n
++cY++"p++baditya@aditya-VirtualBox:~$ ghex
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cbc -d -in encrypted.txt -out decrypted.txt -K 00112233445566778899aabbccddeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat decrypted.txt
Houston do you copy?
This is Confidential information and will be encrypted
Attack at homeland in T-30, prepare NOW!
```

3) CFB

```
rypted.txt -K 00112233445566778899aabbccddeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ ghex
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cfb -d -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cfb -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ ghex encrypted.txt
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-cfb -d -in encrypted.txt -out decrypted.txt -K 00112233445566778899aabbccddeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat decrypted.txt
Houston do you copy?
This is Confidential information and will be encrypted
Attack at homeland in T-30, prepare NOW!
```

4) OFB

```
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ofb -e -in plain.txt -out encrypted.txt -K 00112233445566778899aabbccddeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ ghex encrypted.txt
aditya@aditya-VirtualBox:~$ openssl enc -aes-128-ofb -d -in encrypted.txt -out decrypted.txt -K 00112233445566778899aabbccddeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~$ cat decrypted.txt
Houston do you copy?
This is Confidential information and will be encrypted
Attack at homeland in T-30, prepare NOW!
aditya@aditya-VirtualBox:~$ 
```



Task 4:

```

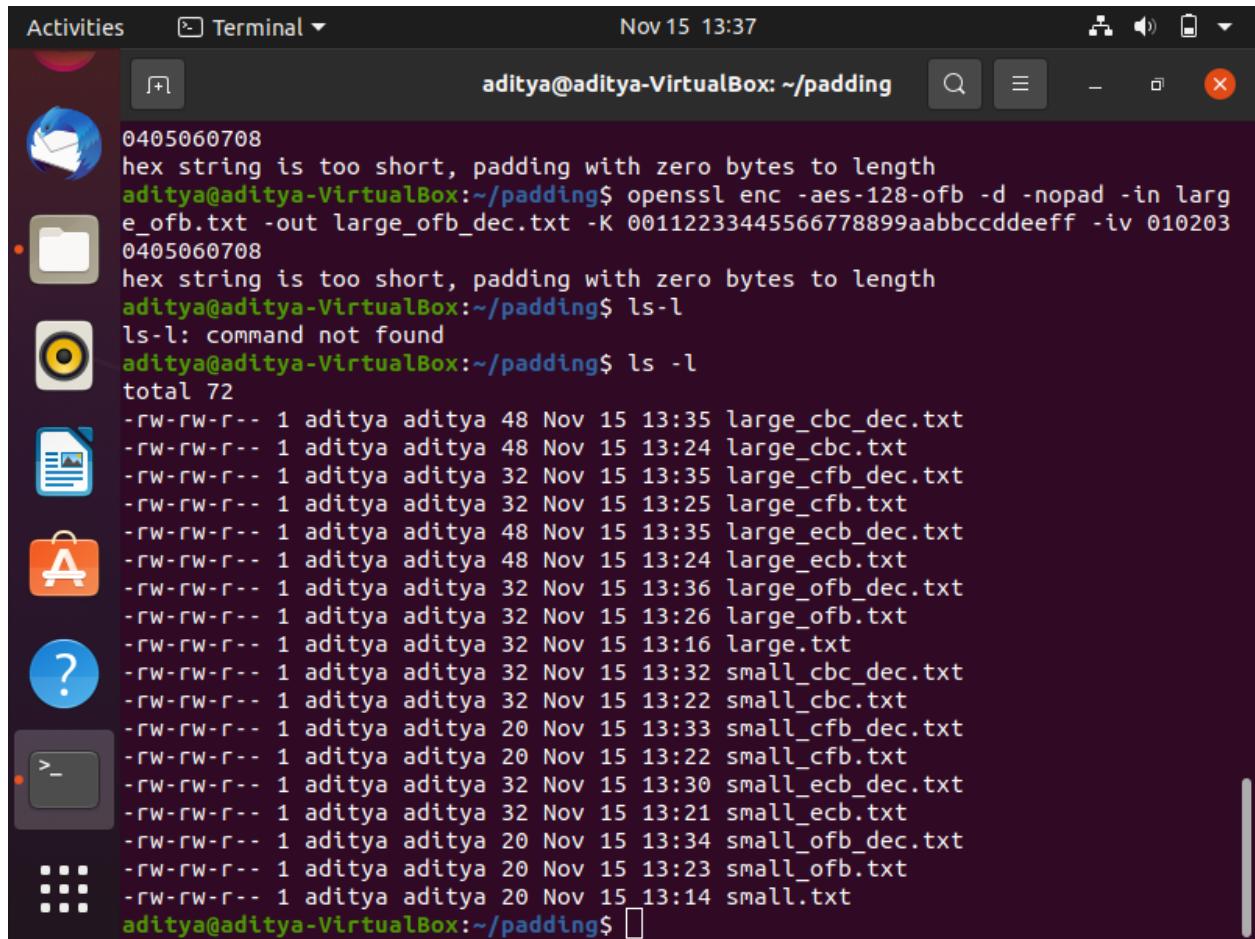
Activities Terminal ▾ Nov 15 13:26
aditya@aditya-VirtualBox: ~/padding
-rw-rw-r-- 1 aditya aditya 31 Nov 15 13:02 large.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:14 small.txt
aditya@aditya-VirtualBox:~/padding$ ls -l
total 8
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:16 large.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:14 small.txt
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -e -in small.txt -
out small_ecb.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -e -in small.txt -
out small_cbc.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -e -in small.txt -
out small_cfb.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -e -in small.txt -
out small_ofb.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -e -in large.txt -
out large_ecb.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -e -in large.txt -
out large_cbc.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -e -in large.txt -
out large_cfb.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -e -in large.txt -
out large_ofb.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ 
```

```
aditya@aditya-VirtualBox:~/padding$ ls -l
total 40
-rw-rw-r-- 1 aditya aditya 48 Nov 15 13:24 large_cbc.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:25 large_cfb.txt
-rw-rw-r-- 1 aditya aditya 48 Nov 15 13:24 large_ecb.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:26 large_ofb.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:16 large.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:22 small_cbc.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:22 small_cfb.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:21 small_ecb.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:23 small_ofb.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:14 small.txt
aditya@aditya-VirtualBox:~/padding$
```

```
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -d -nopad -in small_ecb.txt -out small_ecb_dec.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -d -nopad -in small_cbc.txt -out small_cbc_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -d -nopad -in small_cfb.txt -out small_cfb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -d -nopad -in small_ofb.txt -out small_ofb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -d -nopad -in large_ecb.txt -out large_ecb_dec.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -d -nopad -in large
```

Activities Terminal Nov 15 13:36

```
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -d -nopad -in small_ecb.txt -out small_ecb_dec.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -d -nopad -in small_cbc.txt -out small_cbc_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -d -nopad -in small_cfb.txt -out small_cfb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -d -nopad -in small_ofb.txt -out small_ofb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ecb -d -nopad -in large_ecb.txt -out large_ecb_dec.txt -K 00112233445566778899aabbccddeeff
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cbc -d -nopad -in large_cbc.txt -out large_cbc_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-cfb -d -nopad -in large_cfb.txt -out large_cfb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -d -nopad -in large_ofb.txt -out large_ofb_dec.txt -K 00112233445566778899aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ 
```



A screenshot of a Linux desktop environment, specifically Ubuntu, showing a terminal window. The terminal window is titled "Terminal" and has the command "aditya@aditya-VirtualBox: ~/padding". The terminal shows the following output:

```
0405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ openssl enc -aes-128-ofb -d -nopad -in larg
e_ofb.txt -out large_ofb_dec.txt -K 00112233445566778899aabbcdddeeff -iv 010203
0405060708
hex string is too short, padding with zero bytes to length
aditya@aditya-VirtualBox:~/padding$ ls -l
ls-l: command not found
aditya@aditya-VirtualBox:~/padding$ ls -l
total 72
-rw-rw-r-- 1 aditya aditya 48 Nov 15 13:35 large_cbc_dec.txt
-rw-rw-r-- 1 aditya aditya 48 Nov 15 13:24 large_cbc.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:35 large_cfb_dec.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:25 large_cfb.txt
-rw-rw-r-- 1 aditya aditya 48 Nov 15 13:35 large_ecb_dec.txt
-rw-rw-r-- 1 aditya aditya 48 Nov 15 13:24 large_ecb.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:36 large_ofb_dec.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:26 large_ofb.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:16 large.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:32 small_cbc_dec.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:22 small_cbc.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:33 small_cfb_dec.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:22 small_cfb.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:30 small_ecb_dec.txt
-rw-rw-r-- 1 aditya aditya 32 Nov 15 13:21 small_ecb.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:34 small_ofb_dec.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:23 small_ofb.txt
-rw-rw-r-- 1 aditya aditya 20 Nov 15 13:14 small.txt
aditya@aditya-VirtualBox:~/padding$
```

Task 5:

1) Code:

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
plain_text_inbytes = b"This is a top secret."
cipher_hex =
"8d20e5056a8d24d0462ce74e4904c1b513e10d1df4a2ef2ad4540fae1ca0aaaf9"
answer=""
file = open('words.txt', 'r')
raw_text = file.readlines()
words = [str.strip(line) for line in raw_text]
# print(plain_text_inbytes,cipher_hex)
# print(raw_text)
# print(words)
```

```

for word in words:
    if len(word) >= 16:
        continue
    word = word.lower()
    key = word.encode() + b' ' * (16 - len(word))
    cipher = AES.new(key, AES.MODE_CBC, iv=bytes.fromhex('0' * 32))
    ciphertext = cipher.encrypt(pad(plain_text_inbytes, AES.block_size))
    if bytes.hex(ciphertext) == cipher_hex:
        answer=word
        break
print("Key used to encrypt given plain text:",answer)

```

2) Output:

```

PS C:\Users\adity\Desktop\ADM\College\Coding\Cryptography SS> c;; cd 'c:\Users\adity\Desktop\ADM\College\Co
ding\Cryptography SS'; & 'c:\Users\adity\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\adity\
.vscode\extensions\ms-python.python-2021.11.1422169775\pythonFiles\lib\python\debugpy\launcher' '54708' '--'
'c:\Users\adity\Desktop\ADM\College\Coding\Cryptography SS\task5.py'
Key used to encrypt given plain text: median
-
```

Conclusion:

1) Task 1:

In this task I learned how to use Openssl to encrypt and decrypt messages using various algorithms and modes. I also learned that AES, DES, and CAMELLIA are symmetric key block algorithms that may use the same keys to encrypt and decrypt the data.

2) Task 2:

- a) The encrypted image produced by ECB is very similar to the original and is very intuitive to guess what is the original image by just looking at it.
- b) CBC on the other hand encrypts the image very well, and one cannot make out what the original image is. Hence we understand that CBC is much better for image encryption than ECB.

3) Task 3:

- a) In ECB mode, only one block is affected when a ciphertext problem occurs. As it is decrypted individually. Hence, the corrupted bit in the 30th byte affects the entire block that it is present in.
- b) CBC on the other hand depends on the results of XOR of the previous blocks, hence two blocks are affected.
- c) In CFB mode, there is a problem with the number of n / r blocks.
 r = The number of bits taken at one time to XOR.

- d) In OFB mode, one digit of the 30th byte is corrupted, and in plain text only that byte or that character is broken. Therefore, only OFB mode shows the most promising results and almost all the text will be restored.

4) Task 4:

- a) I observed the size of CBC and ECB encrypted files with the nopad option is 12 bytes more for the 20 bytes file and 16 bytes larger for the 32 bytes file, however the size of OFB and CFB decrypted files is the same.
- b) Hence, we can conclude that ECB and CBC are block ciphers while CFB and OFB are stream ciphers.

5) Task 5:

In this task we understand that with plain text, cipher text, iv and a reasonably large list of possible keys we can find the key by brute force method. Although if the list of possible keys is very large then it is not computationally feasible to do the same. We also learned how to use the pycryptodome library in python to encrypt and decrypt messages.