Noah Cunningham
Reese Pearsall
CSCI 476
11/18/22

Secret-Key Encryption Lab

Task 1: I cracked the cipher, its pretty once you just start railing off correct guesses

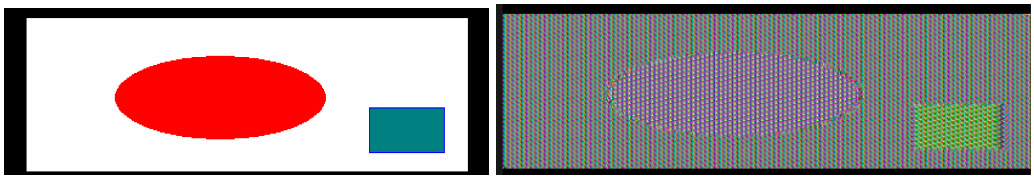
```
[11/19/22]seed@VM:~/.../06_ske$ tr 'yzlxmfeqivbwgcasrnph' 'EHANDISOXWMRCPGFBLYV' < ciphertext.txt >
 out.txt
[11/19/22]seed@VM:~/.../06_ske$ vi out.txt
[11/19/22]seed@VM:~/.../06_ske$ cat out.txt
tHE MACARONI PENGuIN (EuDYPtES CHRYSOLOPHuS) IS A SPECIES OF PENGuIN FOuND FROM tHE SuBANtARCtIC tO
 tHE ANtARCtIC PENINSuLA. ONE OF SIX SPECIES OF CREStED PENGuIN, It IS VERY CLOSELY RELAtED tO tHE
ROYAL PENGuIN, AND SOME AutHORItIES CONSIDER tHE tWO tO BE A SINGLE SPECIES. It BEARS A DIStINCtIVE
 YELLOW CRESt, AND tHE FACE AND uPPERPARtS ARE BLACk AND SHARPLY DELINEAtED FROM tHE WHItE uNDERPAR
tS. ADuLtS WEIGH ON AVERAGE tWELVE POuNDS AND ARE tWENtY EIGHt INCHES IN LENGtH. tHE MALE AND FEMAL
E ARE SIMILAR IN APPEARANCE; tHE MALE IS SLIGHtLY LARGER AND StRONGER WItH A RELAtIVELY LARGER BILL
. LIkE ALL PENGuINS, It IS FLIGHtLESS, WItH A StREAMLINED BODY AND WINGS StIFFENED AND FLAttENED IN
tO FLIPPERS FOR A MARINE LIFEStYLE.[11/19/22]seed@VM:~/.../06_ske$
```

Task 2: I used **-aes-128-ecb**, **-aes-128-cbc** and **-aes-128-cfb** cipher types.

```
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -e -in sometext.txt -
out sometext_out.txt -K 00112233445566778899AABBCCDDEEFF
[11/19/22]seed@VM:~/.../06_ske$ ls
1grams.csv  ciphertext.txt   out.txt           sometext_out.txt
2grams.csv  freq_counter.py  pic_original.bmp  sometext.txt
3grams.csv  out2.txt         sample_xor.py     words.txt
[11/19/22]seed@VM:~/.../06_ske$ cat sometext_out.txt
N??C??Ov???J?K????燈uCM?aB^???))2??{?C?i8ŃU?oNV(S?єC?   ~?,|???>?Q6?#??^/Ě??^?
??J?Kz?Xm????(?)?d????a??????B???w-????!??"?B?"P?
                                                  ???¹Xya?[11/19/22]seed@VM:~/.

[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -e -in sometext.txt -
out sometext_out2.txt -K 00112233445566778899AABBCCDDEEFF -iv 0011223344556677
8899AABBCCDDEEFF
[11/19/22]seed@VM:~/.../06_ske$ cat sometext_out2.txt
??嚨!wX??@e?oT?+??93??4J8?x??T??0_?;?F ?=r?S?I?a:%?????#8??,GJ?o??s???55?"~??L
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -e -in sometext.txt -
out sometext_out3.txt -K 00112233445566778899AABBCCDDEEFF -iv 0011223344556677
3899AABBCCDDEEFF
[11/19/22]seed@VM:~/.../06_ske$ cat sometext_out3.txt
?O?y????JX?#ySd??_??-G?U^?&?e??F35?C??Î?T?
                                            B;
?}+??%?????9??k?-?1??+!H?(???#??????????Ř8Vkn?5???N?F?_? ?"??    ?"
                                                                      ??
                                                                        ?h??c?E  ?
?.??R?Q[11/19/22]seed@VM:~/.../06_ske$
```

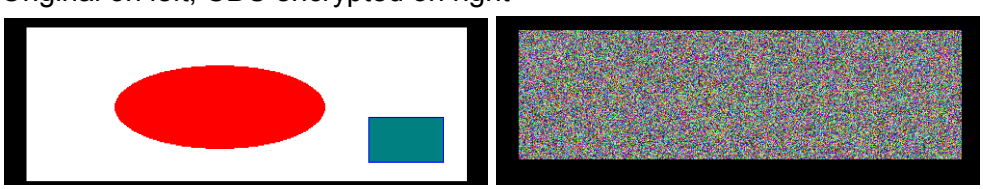
Task 3.1: ECB Original on left, encrypted on right



```
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -e -in pic_original.b
mp -out enc_img.bmp -K 00112233445566778899AABBCCDDEEFF
[11/19/22]seed@VM:~/.../06_ske$ ls
1grams.csv      enc_img.bmp      pic_original.bmp   sometext_out.txt
2grams.csv      freq_counter.py  sample_xor.py      sometext.txt
3grams.csv      out2.txt         sometext_out2.txt  words.txt
ciphertext.txt  out.txt          sometext_out3.txt
[11/19/22]seed@VM:~/.../06_ske$ eog pic_original.bmp
[11/19/22]seed@VM:~/.../06_ske$ head -c 54 pic_original.bmp > header
[11/19/22]seed@VM:~/.../06_ske$ tail -c +55 enc_img.bmp > body
[11/19/22]seed@VM:~/.../06_ske$ cat header body > final_img.bmp
[11/19/22]seed@VM:~/.../06_ske$ eof f
final_img.bmp    freq_counter.py
[11/19/22]seed@VM:~/.../06_ske$ eog final_img.bmp
```

The encrypted image basically looks the same just a little more static with different colors

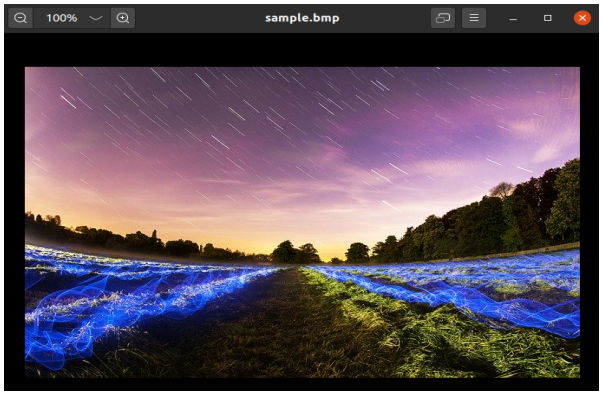
Original on left, CBC encrypted on right



```
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -e -in pic_original.b
mp -out cbc_original.bmp -K 00112233445566778899AABBCCDDEEFF -iv 0001020304050
60708090A0B0C0D0E0F
[11/19/22]seed@VM:~/.../06_ske$ head -c 54 pic_original.bmp > header_org
[11/19/22]seed@VM:~/.../06_ske$ tail -c +55 cbc_original.bmp > body_org
[11/19/22]seed@VM:~/.../06_ske$ cat header_org body_org > final_org.bmp
[11/19/22]seed@VM:~/.../06_ske$ eog final_org.bmp
```

Using cbc encryption seemed to fix the problem and make the image unrecognizable

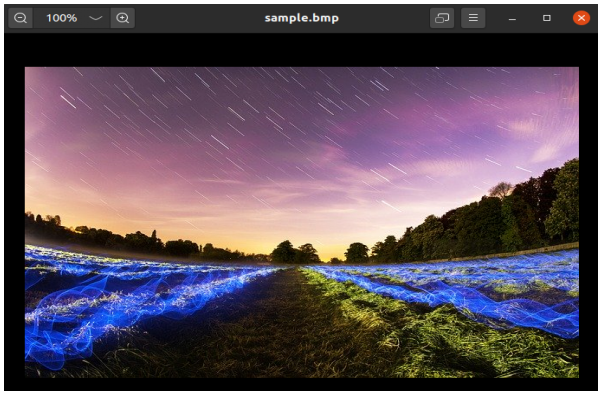
3.2: Original on left, encrypted on right



```
[11/19/22]seed@VM:~/.../06_ske$ eog sample.bmp
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -e -in sample.bmp -ou
t enc_sample.bmp -K 00112233445566778899AABBCCDDEEFF
[11/19/22]seed@VM:~/.../06_ske$ head -c 54 sample.bmp > header
[11/19/22]seed@VM:~/.../06_ske$ tail -c +55 enc_sample.bmp > body
[11/19/22]seed@VM:~/.../06_ske$ cat header body > final_sample.bmp
[11/19/22]seed@VM:~/.../06_ske$ eog final_sample.bmp
```

The encrypted image looks like complete TV static, and can't I make out anything. I think the reason you cant make out anything in the 2nd image is because its more ‘complicated’, has more pixels and colors and less basic shapes.

Original on left, CBC encrypted on right



Practically looks the same as before ecb

Task 4.1:

```
[11/19/22]seed@VM:~/.../06_ske$ echo -n "12345" > f1.txt
[11/19/22]seed@VM:~/.../06_ske$ echo -n "1234567890" > f2.txt
[11/19/22]seed@VM:~/.../06_ske$ echo -n "1234567890123456" > f3.txt
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -e -in f1.txt -out f1
_cbc.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0
E0F
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -e -in f2.txt -out f2
_cbc.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0
E0F
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -e -in f3.txt -out f3
_cbc.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0
E0F
```

The size of each file

```
-rw-rw-r-- 1 seed seed   16 Nov 19 21:17 f1_cbc.txt
-rw-rw-r-- 1 seed seed    5 Nov 19 21:16 f1.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:18 f2_cbc.txt
-rw-rw-r-- 1 seed seed   10 Nov 19 21:16 f2.txt
-rw-rw-r-- 1 seed seed   32 Nov 19 21:18 f3_cbc.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:16 f3.txt
```

Seems like when encrypting it adds the difference bytes to the next ‘byte milestone’ (16, 32, 64…)

Decrypted using -nopad, and you can see all the padding

```
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -d -in  f1_cbc.txt -o
ut f1_dec.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B
0C0D0E0F -nopad
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -d -in  f2_cbc.txt -o
ut f2_dec.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B
0C0D0E0F -nopad
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -d -in  f3_cbc.txt -o
ut f3_dec.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B
0C0D0E0F -nopad
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f1_dec.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b 0b  |12345...........|
00000010
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f2_dec.txt
00000000  31 32 33 34 35 36 37 38  39 30 06 06 06 06 06 06  |1234567890......|
00000010
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f3_dec.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010  10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10  |................|
00000020
```

4.2:

ECB:

```
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -e -in f1.txt -out f1
_ecb.txt -K 00112233445566778899AABBCCDDEEFF
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -e -in f2.txt -out f2
_ecb.txt -K 00112233445566778899AABBCCDDEEFF
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -e -in f3.txt -out f3
_ecb.txt -K 00112233445566778899AABBCCDDEEFF
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -d -in f1_ecb.txt -ou
t f1_ecb_dec.txt -K 00112233445566778899AABBCCDDEEFF -nopad
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -d -in f2_ecb.txt -ou
t f2_ecb_dec.txt -K 00112233445566778899AABBCCDDEEFF -nopad
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -d -in f3_ecb.txt -ou
t f3_ecb_dec.txt -K 00112233445566778899AABBCCDDEEFF -nopad
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f1_ecb_dec.txt
00000000  31 32 33 34 35 0b 0b 0b  0b 0b 0b 0b 0b 0b 0b 0b  |12345...........|
00000010
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f2_ecb_dec.txt
00000000  31 32 33 34 35 36 37 38  39 30 06 06 06 06 06 06  |1234567890......|
00000010
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f3_ecb_dec.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010  10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10  |................|
00000020
```

CFB:Doesnt have padding because you 1/2-byte blocks to decrypted

```
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -e -in f1.txt -out f1_cfb.txt -K 00112
233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -e -in f2.txt -out f2_cfb.txt -K 00112
233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -e -in f3.txt -out f3_cfb.txt -K 00112
233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -d -in  f1_cfb.txt -out f1_dec_cfb.txt
 -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F -nopad
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -d -in  f2_cfb.txt -out f2_dec_cfb.txt
 -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F -nopad
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -d -in  f3_cfb.txt -out f3_dec_cfb.txt
 -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F -nopad
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f1_dec_cfb.txt
00000000  31 32 33 34 35                                       |12345|
00000005
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f2_dec_cfb.txt
00000000  31 32 33 34 35 36 37 38  39 30                       |1234567890|
0000000a
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f3_dec_cfb.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010
```

OFB: Doesnt have padding because you 1/2-byte blocks to decrypted

```
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ofb -e -in f1.txt -out f1_ofb.txt -K 00112
233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ofb -e -in f2.txt -out f2_ofb.txt -K 00112
233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ofb -e -in f3.txt -out f3_ofb.txt -K 00112
233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ofb -d -in f1_ofb.txt -out f1_ofb_dec.txt
-K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F -nopad
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ofb -d -in f2_ofb.txt -out f2_ofb_dec.txt
-K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F -nopad
[11/19/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ofb -d -in f3_ofb.txt -out f3_ofb_dec.txt
-K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F -nopad
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f1_ofb_dec.txt
00000000  31 32 33 34 35                                       |12345|
00000005
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f2_ofb_dec.txt
00000000  31 32 33 34 35 36 37 38  39 30                       |1234567890|
0000000a
[11/19/22]seed@VM:~/.../06_ske$ hexdump -C f3_ofb_dec.txt
00000000  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
00000010
```

Sizes for all the ECB, CFB and OFB Files:

```
seed@VM: ~/.../06_ske
seed@VM: ~/.../06_ske
-rw-rw-r-- 1 seed seed   16 Nov 19 21:17 f1_cbc.txt
-rw-rw-r-- 1 seed seed    5 Nov 19 21:59 f1_cfb.txt
-rw-rw-r-- 1 seed seed    5 Nov 19 22:00 f1_dec_cfb.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:35 f1_dec.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:43 f1_ecb_dec.txt
-rw-rw-r-- 1 seed seed    0 Nov 19 21:54 f1_ecb.txt
-rw-rw-r-- 1 seed seed    5 Nov 19 22:20 f1_ofb_dec.txt
-rw-rw-r-- 1 seed seed    5 Nov 19 22:20 f1_ofb.txt
-rw-rw-r-- 1 seed seed    5 Nov 19 21:16 f1.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:18 f2_cbc.txt
-rw-rw-r-- 1 seed seed   10 Nov 19 21:59 f2_cfb.txt
-rw-rw-r-- 1 seed seed   10 Nov 19 22:01 f2_dec_cfb.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:35 f2_dec.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:43 f2_ecb_dec.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:40 f2_ecb.txt
-rw-rw-r-- 1 seed seed   10 Nov 19 22:21 f2_ofb_dec.txt
-rw-rw-r-- 1 seed seed   10 Nov 19 22:20 f2_ofb.txt
-rw-rw-r-- 1 seed seed   10 Nov 19 21:16 f2.txt
-rw-rw-r-- 1 seed seed   32 Nov 19 21:18 f3_cbc.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:59 f3_cfb.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 22:01 f3_dec_cfb.txt
-rw-rw-r-- 1 seed seed   32 Nov 19 21:35 f3_dec.txt
-rw-rw-r-- 1 seed seed   32 Nov 19 21:43 f3_ecb_dec.txt
-rw-rw-r-- 1 seed seed   32 Nov 19 21:41 f3_ecb.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 22:21 f3_ofb_dec.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 22:20 f3_ofb.txt
-rw-rw-r-- 1 seed seed   16 Nov 19 21:16 f3.txt
```

Task 5.1:

How much information can you recover by decrypting the corrupted file, if the encryption mode is ECB, CBC, CFB, or OFB, respectively?

ECB: You wont beagle to revolver the entire n-byte block where the corrupted byte is located, because you use n-byte blocks to encrypt/decrypt

CBC: You wont beagle to revolver the entire n-byte block where the corrupted byte is located, because you use n-byte blocks to encrypt/decrypt

CFB: I predict you can recover every byte except the corrupted byte, and potentially another byte next to the corrupted byte, because you use 1-2 byte blocks to encrypt/decrypt

OFB: I predict you can recover every byte except the corrupted byte, and potentially another byte next to the corrupted byte, because you use 1-2 byte blocks to encrypt/decrypt

Task 5.2:

```
[11/20/22]seed@VM:~/.../06_ske$ echo "12345678901234567890123456789012345678901234567890123456789012345678
901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
456789012345678901234567890123456789012345678901234567890123456789012345678" > f4.t
xt
```

```
-rw-rw-r-- 1 seed seed 1000 Nov 20 01:19 f4.txt
```

ECB:

```
[11/20/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -e -in f4.txt -out f4_ecb
.txt -K 00112233445566778899AABBCCDDEEFF
```

Edited the value to 00

```
File  Edit  View  Windows  Help
00000000BB A3 01 73 75 E3 06 7D CE 4A 3E EF CD 2E 84 70...su..}.J>....p
0000001094 FF 08 AD 95 0E CA 0D 1D 5B 65 65 0C 30 72 AB.........[ee.0r.
0000002022 F9 F5 40 B5 41 82 B8 1A 07 A7 B2 ED DD D2 57"..@.A........W
00000030F8 DC E1 15 CF 10 00 02 73 1D 14 59 77 B8 26 F9.......s..Yw.&.
00000040BC 02 9B 10 0D A2 B1 88 A2 3B 67 5A 08 37 28 07.........;gZ.7(.
00000050BB A3 01 73 75 E3 06 7D CE 4A 3E EF CD 2E 84 70...su..}.J>....p
0000006094 FF 08 AD 95 0E CA 0D 1D 5B 65 65 0C 30 72 AB.........[ee.0r.
0000007022 F9 F5 40 B5 41 82 B8 1A 07 A7 B2 ED DD D2 57"..@.A........W
00000080F8 DC E1 15 CF 10 9D 02 73 1D 14 59 77 B8 26 F9........s..Yw.&.
00000090BC 02 9B 10 0D A2 B1 88 A2 3B 67 5A 08 37 28 07.........;gZ.7(.
```

```
[11/20/22]seed@VM:~/.../06_ske$ ghex f4_ecb.txt
[11/20/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-ecb -d -in f4_ecb.txt -out f4
_ecb_dec.txt -K 00112233445566778899AABBCCDDEEFF
[11/20/22]seed@VM:~/.../06_ske$ cat f4_ecb_dec.txt
1234567890123456789012345678901234567890123456789012345678)~????9??/5678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456789012345678
9012345678901234567890123456789012345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456789012345678
9012345678901234567890123456789012345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456789012345678
901234567
[11/20/22]seed@VM:~/.../06_ske$
```

Appears i was able to recover most of the data, 16 bytes seem to be missing

Task 5.3:

```
f4_cbc.txt - GHex
File  Edit  View  Windows  Help
0000000034 FE 4A 82 B1 62 4A B7 58 AB 27 6B 1D 93 F0 064.J..bJ.X.'k....
00000010B0 EE A3 A3 DD 86 84 AE 5A A4 4E 64 28 45 A1 CA........Z.Nd(E..
00000200A CD 7D F7 93 AD E7 F2 98 8F 73 85 50 74 13 96..}.......s.Pt..
00000030F5 81 7D 42 89 EC 23 00 79 2A 77 9A 12 44 B7 EB..}B..#.y*w..D..
000000401A 99 E2 03 0B 63 28 FB 65 87 BC 5C E0 1D 68 ED.....c(.e..\..h.
00000050C2 CB B9 7B 29 B5 95 2F 3B B4 A7 01 E7 04 59 D0...{)../;.....Y.
0000006007 10 A6 D3 EE 21 73 C5 1F 19 EE 3B 13 3D B5 31.....!s....;.=.1
0000007029 94 B8 31 60 72 7A 67 17 05 0C 11 10 F0 72 23)..1`rzg......r#
0000008067 10 D8 7E B8 9F 6E 8B 77 C6 EB 44 01 C1 E4 38g..~..n.w..D...8
0000009062 4B 3C A8 FB 20 97 1A FD 39 78 92 67 BF DA F2bK<.. ...9x.g...
```

```
[11/20/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -e -in f4.txt -out f4_cbc
.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F
[11/20/22]seed@VM:~/.../06_ske$ ghex f4_cbc.txt
[11/20/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cbc -d -in f4_cbc.txt -out f4
_cbc_dec.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0
E0F
[11/20/22]seed@VM:~/.../06_ske$ cat f4_cbc_dec.txt
1234567890123456789012345678901234567890123456789?????+N??        ?*?5678901m3456789
0123456789012345678901234567890123456789012345678901234567890123456789012345678901
2345678901234567890123456789012345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890123456789012345678901234567890123456789012345
6789012345678901234567890123456789012345678901234567890123456789012345678901234567
8901234567890123456789012345678901234567890123456789012345678901234567890123456789
0123456789012345678901234567890123456789012345678901234567890123456789012345678901
2345678901234567890123456789012345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890123456789012345678901234567890123456789012345
6789012345678901234567890123456789012345678901234567890123456789012345678901234567
8901234567890123456789012345678901234567890123456789012345678901234567890123456789
0123456789012345678901234567890123456789012345678901234567890123456789012345678901
2345678901234567
[11/20/22]seed@VM:~/.../06_ske$
```

Appears i was able to recover most of the data, 16 bytes seem to be missing, but also there is a character that is different close by, that me is supposed to be a 2, so i guess 17 bytes are missing

Task 5.4

```
File  Edit  View  Windows  Help
0000000016 AD 84 7E 40 44 24 66 B6 AB BF C4 E2 DA D5 35...~@D$f.......5
0000001056 71 55 F0 75 42 36 23 F0 CE 71 27 D2 12 54 00VqU.uB6#..q'..T.
00000020FD 28 F7 2A 1C 85 4B E5 93 05 4A C0 3C 66 8C F7.(.*..K...J.<f..
0000003054 B4 A2 3C 5F BC 0? 90 47 36 30 22 BD 94 5B 02T..<_..?.G60"..[.
00000040DF E0 90 10 6C 7C 33 A2 AC 3F EF B4 09 CA 17 A5....l|3..?......
00000050B4 92 BB D1 9C 0D 86 4B D4 E7 72 35 A0 1F 7C 2B.......K..r5..|+
0000006084 73 07 E4 5C A5 B0 75 AA 8C A4 C5 65 67 9C 59.s..\..u....eg.Y
00000070FF 94 AC F3 CF 3C 2F 3F F9 9B B3 41 B9 4B 3C 72.....</?...A.K<r
0000008016 E9 2A 0F A4 19 62 74 C5 7F 03 03 D4 94 4B 6E..*...bt......Kn
000000900F 06 FB 01 7E 42 2C 46 9E B6 5C BB B2 C2 10 28....~B,F..\....(
```

```
[11/20/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -e -in f4.txt -out f4_cfb
.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0E0F
[11/20/22]seed@VM:~/.../06_ske$ ghex f4_cfb.txt
[11/20/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -d -in f4_cfb.txt -out f4
_cfb_dec.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0D0
E0F
[11/20/22]seed@VM:~/.../06_ske$ cat f4_cfb_dec.txt
1234567890123456789012345678901234567890123456789012340678901234006hU*0} 000Q123456
7890123456789012345678901234567890123456789012345678901234567890123456789012345678
9012345678901234567890123456789012345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456789012345678
9012345678901234567890123456789012345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890123456789012345678901234567890123456789012345678901234
5678901234567890123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456789012345678
9012345678
```

Similar to 5.3 16 bytes seem to be missing in one chunk but there is another byte to the left of that chunk where its supposed to be a 5 but is a ?

Task 6.1

When i used the same -iv both encrypted files are the same

```
[11/20/22]seed@VM:~/.../06_ske$ echo "Hello, my name is Noah" > task6_1.txt
[11/20/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -e -in task6_1.txt -out t
ask6_1_enc.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C0
D0E0F
[11/20/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -e -in task6_1.txt -out t
ask6_1_enc2.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C
0D0E0F

[11/20/22]seed@VM:~/.../06_ske$ diff task6_1_enc.txt task6_1_enc2.txt
[11/20/22]seed@VM:~/.../06_ske$
```

When i used different -iv the encrypted files are different

```
[11/20/22]seed@VM:~/.../06_ske$ openssl enc -aes-128-cfb -e -in task6_1.txt -out t
ask6_1_enc3.txt -K 00112233445566778899AABBCCDDEEFF -iv 000102030405060708090A0B0C
0D0E00
[11/20/22]seed@VM:~/.../06_ske$ diff task6_1_enc.txt task6_1_enc3.txt
1c1
< o??&?^33???????j????$X
\ No newline at end of file
---
> ???? ??
         ???L?,?

\ No newline at end of file
[11/20/22]seed@VM:~/.../06_ske$
```

Task 6.2

Plaintext (P1): This is a known message!
Ciphertext (C1): a469b1c502c1cab966965e50425438e1bb1b5f9037a4c159

Plaintext (P2): (unknown to you)
Ciphertext (C2): bf73bcd3509299d566c35b5d450337e1bb175f903fafc159

```
[11/20/22]seed@VM:~/.../06_ske$ python3 sample_xor.py
f001d8b622a8b99907b6353e2d2356c1d67e2ce356c3a478
1b1a0d165253536c0055050d07570f00000c0000080b0000
00000000000000000000000000000000000000000000000000
```

The encryption string is: f001d8b622a8b99907b6353e2d2356c1d67e2ce356c3a478

So we should be able to use that string XOR on C2 in order to get the answer

Edited file to XOR encryption string and C2 and then converted the output of that to a string

```
# XOR strings (ascii strings and hex strings).
#

MSG   = "This is a known message!"
HEX_1 = "a469b1c502c1cab966965e50425438e1bb1b5f9037a4c159"
HEX_2 = "bf73bcd3509299d566c35b5d450337e1bb175f903fafc159"
HEX_3 = "f001d8b622a8b99907b6353e2d2356c1d67e2ce356c3a478"

# XOR two bytearrays
def xor(first, second):
   return bytearray(x^y for x,y in zip(first, second))

D1 = bytes(MSG, 'utf-8')      # Convert ascii string to bytearray
D2 = bytearray.fromhex(HEX_1) # Convert hex string to bytearray
D3 = bytearray.fromhex(HEX_2) # Convert hex string to bytearray
D4 = bytearray.fromhex(HEX_3)

r1 = xor(D1, D2)
r2 = xor(D2, D3)
r3 = xor(D2, D2)
r4 = xor(D3, D4)
print(r1.hex())
print(r2.hex())
print(r3.hex())
print(r4.hex())
print(r4.decode('utf8'))
"sample_xor.py" 29L, 768C                                    29,24         Bot
```

Output

```
[11/20/22]seed@VM:~/.../06_ske$ python3 sample_xor.py
f001d8b622a8b99907b6353e2d2356c1d67e2ce356c3a478
1b1a0d165253536c0055050d07570f00000c0000080b0000
00000000000000000000000000000000000000000000000000
4f726465723a204c61756e63682061206d697373696c6521
Order: Launch a missile!
```