

Lab 2, Part 3

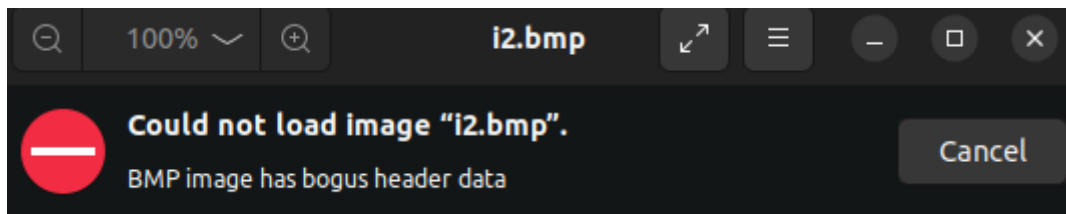
3.1

- use the openssl to encrypt the picture

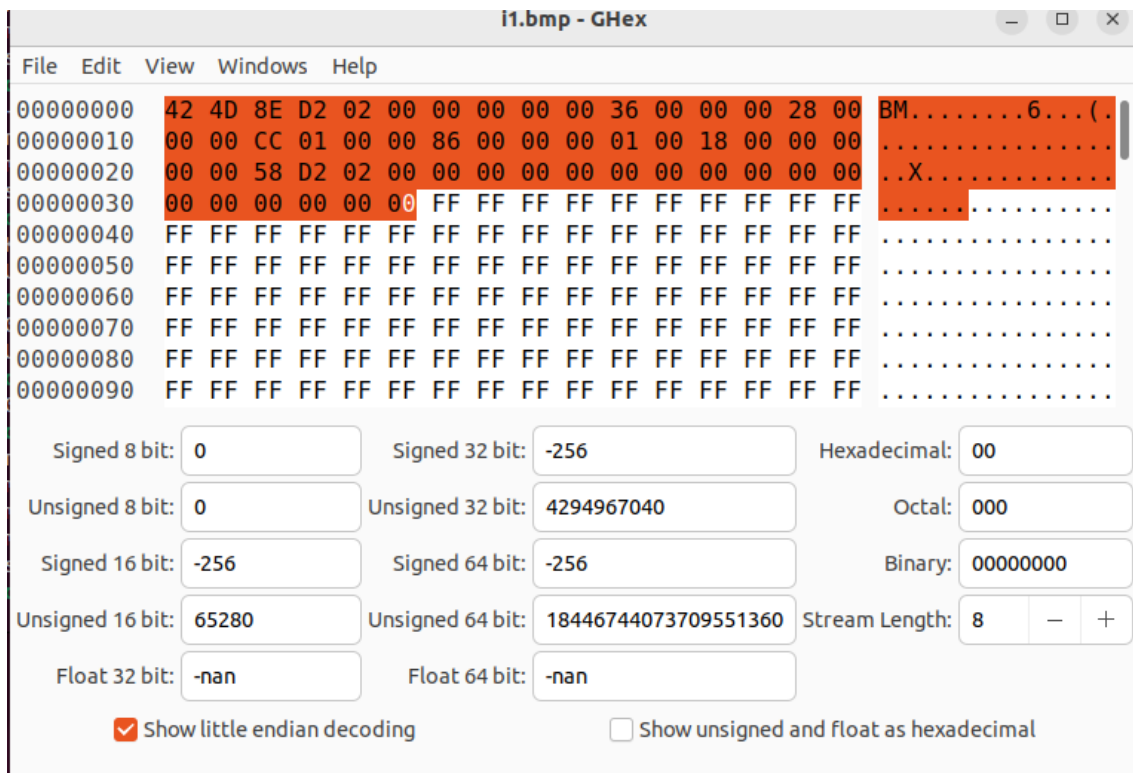
- ```
openssl aes-128-ecb -e -in i1.bmp -out i2.bmp -K 00112233445566778889aabbccddeeff
```

```
ddd@ddd-virtual-machine:~/Desktop$ openssl aes-128-ecb -e -in i1.bmp -out i2.bmp -K 00112233445566778889aabbccddeeff
```

- As the bits of the 2.bmp as been modified (the bits at the beginning of the file to prove the project we can not open the picture at once.



- use ghex to check i1.bmp file in picture we can see what's the first few bits in the .bmp file.



- 打开i2.bmp文件

i2.bmp - GHex

File Edit View Windows Help

|          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |                |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----------------|
| 00000000 | F4 | 21 | 1C | BB | 17 | F2 | 8D | 58 | 16 | 87 | A2 | 40 | D8 | 09 | 9D | FF | !                 | .....X...@.... |
| 00000010 | 6F | 4C | 32 | F8 | 47 | F2 | 47 | 8D | 6C | 01 | DF | B9 | E0 | C4 | 01 | 84 | oL2.G.G.l.....    |                |
| 00000020 | 66 | B8 | BC | D0 | B6 | 3E | 6B | 30 | 78 | A0 | C7 | CD | 8C | FF | 8F | D9 | f....>k0x.....    |                |
| 00000030 | 40 | BD | 1E | F5 | F5 | 27 | 07 | AA | B0 | 90 | 5B | 50 | FB | 5E | F8 | E6 | @....'.....[P.^.. |                |
| 00000040 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<.....        |                |
| 00000050 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<.....        |                |
| 00000060 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<.....        |                |
| 00000070 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<.....        |                |
| 00000080 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<.....        |                |
| 00000090 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<.....        |                |

Signed 8 bit: -12

Signed 32 bit: -1155784204

Hexadecimal: F4

Unsigned 8 bit: 244

Unsigned 32 bit: 3139183092

Octal: 364

Signed 16 bit: 8692

Signed 64 bit: 6381022430791213556

Binary: 11110100

Unsigned 16 bit: 8692

Unsigned 64 bit: 6381022430791213556

Stream Length: 8 - +

Float 32 bit: -2.382395e-03

Float 64 bit: 3.775743e+118

☒ Show little endian decoding
 ☐ Show unsigned and float as hexadecimal

i2.bmp - GHex

File Edit View Windows Help

|          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |               |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|---------------|
| 00000000 | 42 | 4D | 8E | D2 | 02 | 00 | 00 | 00 | 00 | 00 | 36 | 00 | 00 | 00 | 28 | 00 | B          | M.....6....(. |
| 00000010 | 00 | 00 | CC | 01 | 00 | 00 | 86 | 00 | 00 | 00 | 01 | 00 | 18 | 00 | 00 | 00 |            | .....         |
| 00000020 | 00 | 00 | 58 | D2 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |            | ..X.....      |
| 00000030 | 00 | 00 | 00 | 00 | 00 | 00 | 07 | AA | B0 | 90 | 5B | 50 | FB | 5E | F8 | E6 |            | .....[P.^..   |
| 00000040 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<..... |               |
| 00000050 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<..... |               |
| 00000060 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<..... |               |
| 00000070 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<..... |               |
| 00000080 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<..... |               |
| 00000090 | 47 | C9 | 96 | BB | 3C | 11 | C7 | 0A | BC | F0 | 81 | BF | F0 | 99 | 9A | B8 | G...<..... |               |

Signed 8 bit: 66

Signed 32 bit: -762426046

Hexadecimal: 42

Unsigned 8 bit: 66

Unsigned 32 bit: 3532541250

Octal: 102

Signed 16 bit: 19778

Signed 64 bit: 12122475842

Binary: 01000010

Unsigned 16 bit: 19778

Unsigned 64 bit: 12122475842

Stream Length: 8 - +

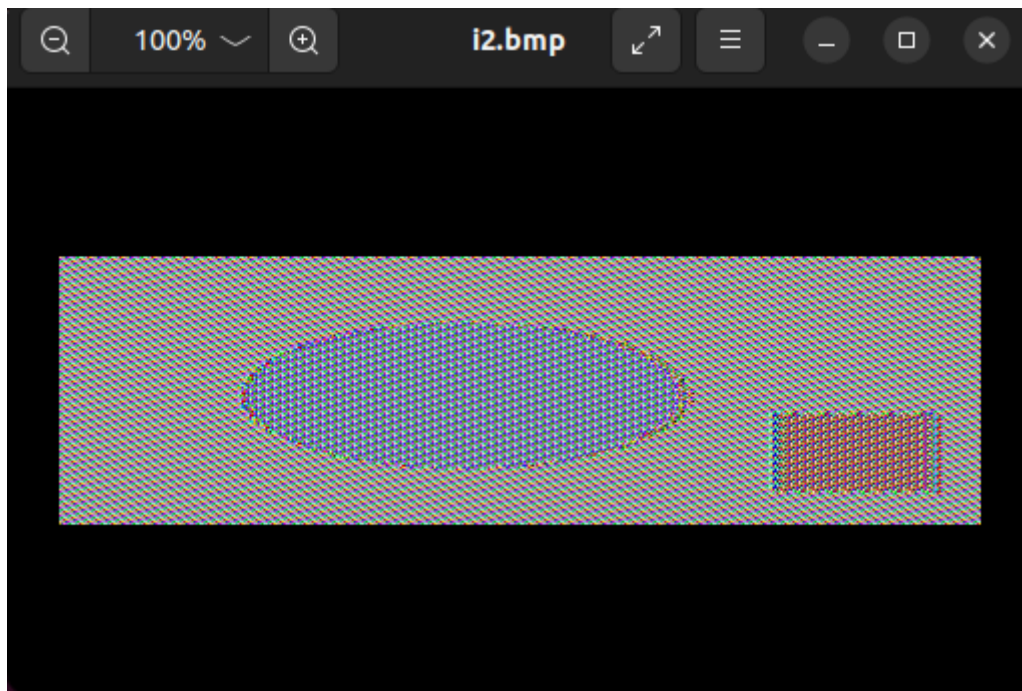
Float 32 bit: -3.055908e+11

Float 64 bit: 5.989299e-314

☒ Show little endian decoding
 ☐ Show unsigned and float as hexadecimal

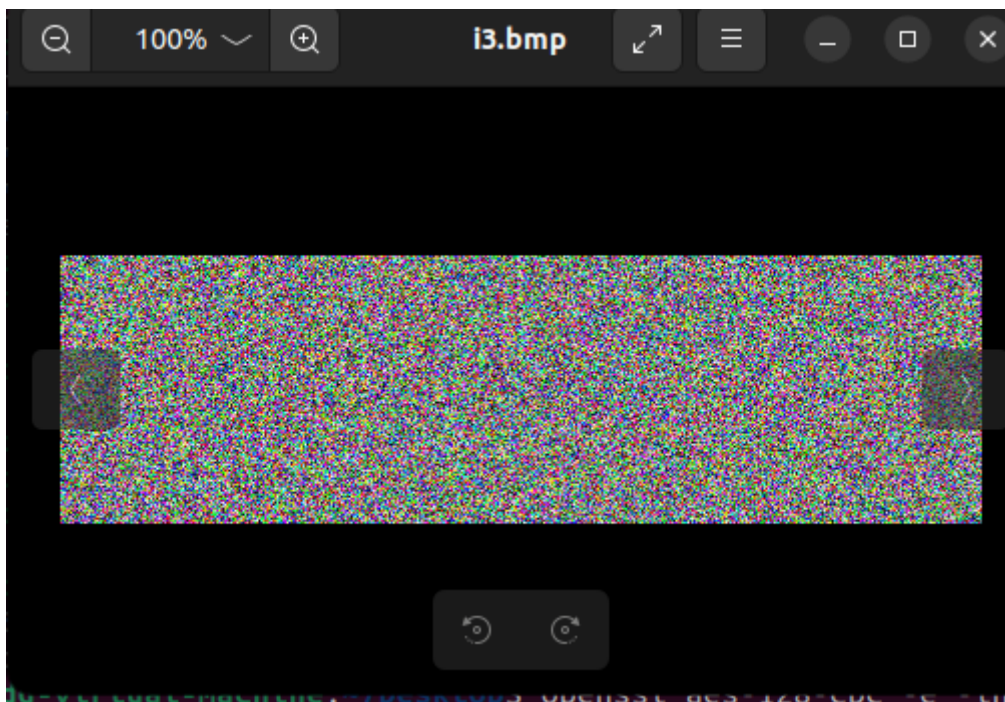
Offset: 0x0

- after replacing the head we can see i2 as the encrypted message.



- same option to generate i3(in cbc)

```
ddd@ddd-virtual-machine:~/Desktop$ openssl aes-128-cbc -e -in i1.bmp -out i3.bmp
-K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
ddd@ddd-virtual-machine:~/Desktop$ ghex i3.bmp
```



- it is clear that cbc can hide the information of the picture much better.

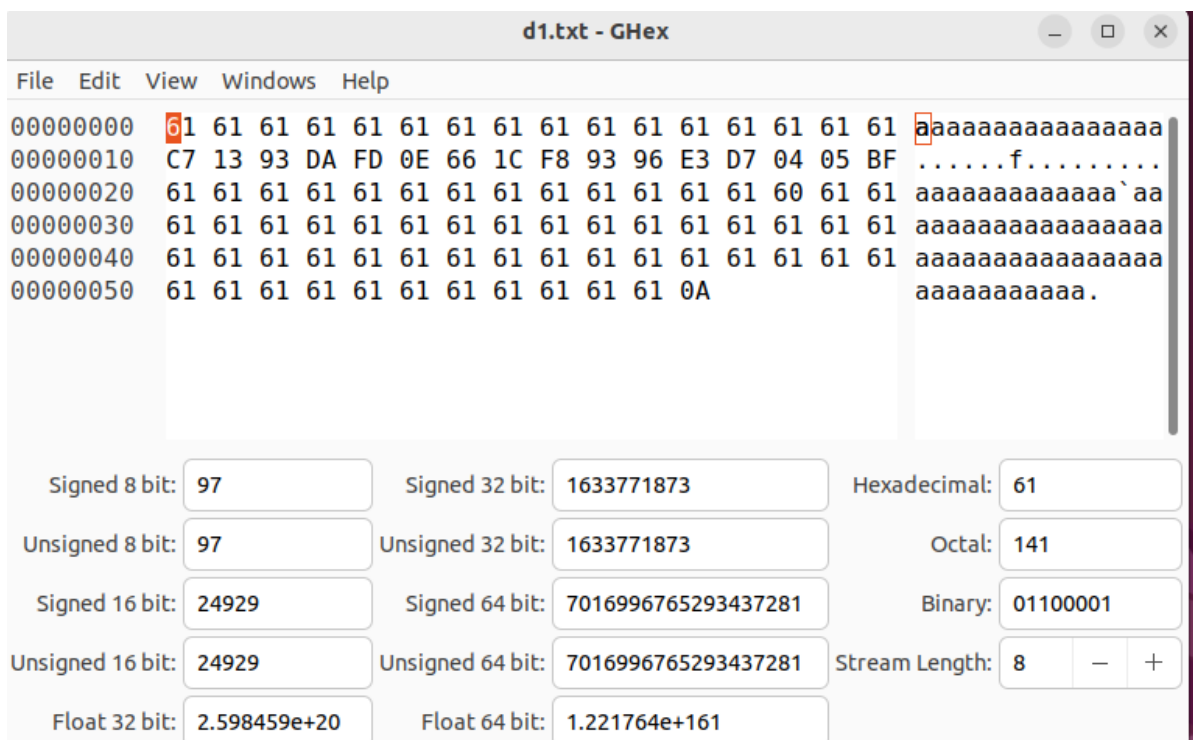
## 3.2

### CBC

- create m.txt to be encrypted



- The result of CBC decrypted after modify the c



- cause every 64bits are a group and cbc use xor between 2 groups,so the group2 and first 7 bytes of group3 are broken.

## ECB

```
ddd@ddd-virtual-machine:~/Desktop$ openssl aes-128-ecb -e -in m.txt -out c1.txt
-K 00112233445566778889aabbccddeeff
ddd@ddd-virtual-machine:~/Desktop$ ghex c1.txt
ddd@ddd-virtual-machine:~/Desktop$ openssl aes-128-ecb -d -in c1.txt -out d2.txt
-K 00112233445566778889aabbccddeeff
ddd@ddd-virtual-machine:~/Desktop$ ghex d2.txt
```



```

d2.txt - GHex
File Edit View Windows Help
00000000 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa
00000010 A9 29 33 9B 93 30 3D 4A 7A E1 2D 7F 4D 5B 54 24 .)3..0=Jz.-.M[T$
00000020 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa
00000030 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa
00000040 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa
00000050 61 61 61 61 61 61 61 61 61 61 61 0A aaaaaaaaaa.

```

- only group 2 are broken, cause ecb do not have any relationship between each group.

## CFB

- option:

```

ddd@ddd-virtual-machine:~/Desktop$ openssl aes-128-cfb -e -in m.txt -out c3.txt
-K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
ddd@ddd-virtual-machine:~/Desktop$ gehx c3
Command 'gehx' not found, did you mean:
 command 'genx' from snap genx (v3.6.20-2-gc33f2ab)
 command 'ghex' from deb ghex (3.41.1-1)
 command 'genx' from deb python3-genx (3.0.2-1)
See 'snap info <snapname>' for additional versions.
ddd@ddd-virtual-machine:~/Desktop$ gehx c3.txt
Command 'gehx' not found, did you mean:
 command 'genx' from snap genx (v3.6.20-2-gc33f2ab)
 command 'ghex' from deb ghex (3.41.1-1)
 command 'genx' from deb python3-genx (3.0.2-1)
See 'snap info <snapname>' for additional versions.
ddd@ddd-virtual-machine:~/Desktop$ ghex c3.txt
ddd@ddd-virtual-machine:~/Desktop$ ghex c3.txt
ddd@ddd-virtual-machine:~/Desktop$ openssl aes-128-cfb -d -in c3.txt -out d3.txt
-K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
ddd@ddd-virtual-machine:~/Desktop$ ghex d3.txt

```

- result

```

d3.txt - GHex
File Edit View Windows Help
00000000 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa
00000010 61 61 61 61 61 61 61 61 61 61 61 71 61 61 61 61 aaaaaaaaaaaaaqaaaa
00000020 E0 72 82 6B B9 AB E0 34 63 3B AA FE B6 EC FD 40 .r.k...4c;.....@
00000030 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa
00000040 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa
00000050 61 61 61 61 61 61 61 61 61 61 61 0A aaaaaaaaaa.

```

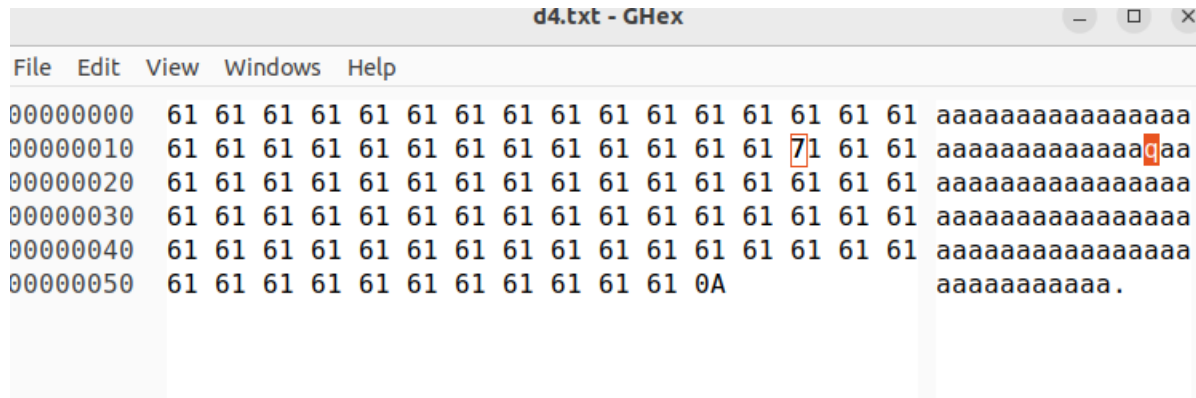
- conclusion: the wrong bit gose into the register and do xor option with next group so the group3's information is wrong.

# OFB

- option

```
ddd@ddd-virtual-machine:~/Desktop$ openssl aes-128-ofb -e -in m.txt -out c4.txt
-K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
ddd@ddd-virtual-machine:~/Desktop$ ghex c4.txt
ddd@ddd-virtual-machine:~/Desktop$ openssl aes-128-ofb -d -in c4.txt -out d4.txt
-K 00112233445566778889aabbccddeeff -iv 0102030405060708
hex string is too short, padding with zero bytes to length
ddd@ddd-virtual-machine:~/Desktop$ ghex d4.txt
```

- result



- conclusion: ofb dose not have the problem of diffusion, besides it uses flow to encrypted so only byte no.30 is affected.