Saishnu Ramesh Kumar (300758706)

CSCI 191T (Assignment 4) – XSS, Secret Key Encryption, and One-Way Hash Functions

<u>Section 1:</u>

1) Potentially considering if the logic is perfect, yes. The main cause of XSS attacks is when the attacker embeds malicious code, mainly JavaScript, into the victim's browser, therefore developing a filtering system can help prevent any JavaScript code from being both uploaded and downloaded.

2) The CSP is effective in defeating Cross-Site Scripting attacks because it prevents malicious code from being injected into a site. It can prevent these forms of threats because it only limits the sources of content that is allowed to be displayed. The downside of this approach is that it can block actual safe scripts from running which are not owned by the CSP and may cause the site to not work properly.

3) The padding data will be 77777. If the message has 64 bytes, it will be easier to run the program because the 64 bytes padding memory will be increased and it will allow the message to run faster and it is the more standard for the system to compute.

4) Bob will still be able to receive the entire plaintext because the plaintext in both the OFB and CTR models do not interfere with the encryption mode and in both instances, they are just added to the XOR statement to then obtain the ciphertext. If it

was on the decryption mode, Bob will be able to recover all the bits except for the 2nd bit and this applies to both models as well.

5) For a one-way hash function, this function is not a good method because it will be difficult to find x if the database becomes big, in this case out of 10000 and it may also cause possible collision among data, hence having a collision resistant to reduce/prevent that.

6) Yes, it is better to save salt in the plaintext because the salt is already provided with security as it must be added with plaintext. They also are easier to save, and it will be a more difficult time for the attack to breach its security. Moreover, when using salt, the same input results in different hashes.

7) Yes, I do agree that it can improve the security because when the hash is sent, it is more encrypted and protected from an attacker potential stealing that data. For password verification, the user would need to tell a secret password which cannot be stored using plaintext. If the user were to login to their profile with their password, it would need to verify itself with the password that is stored on the server to see if they match. Nonetheless, the more secure way is by using salt as it ensures that the hashes are always distinct from one another.

8) In Linux, the Operating System stores passwords in the /etc/shadow file. Within the shadow file, the passwords stored have 3 parts: the algorithm used, salt, and password hash. The salt and password hash are encoded as printable characters whereas the

algorithm is not. The reasoning as to why Linus applies a password hash for many rounds, like 5000, is because it can slow down any potential brute force attacks and keeps the passwords secured from any attacker.

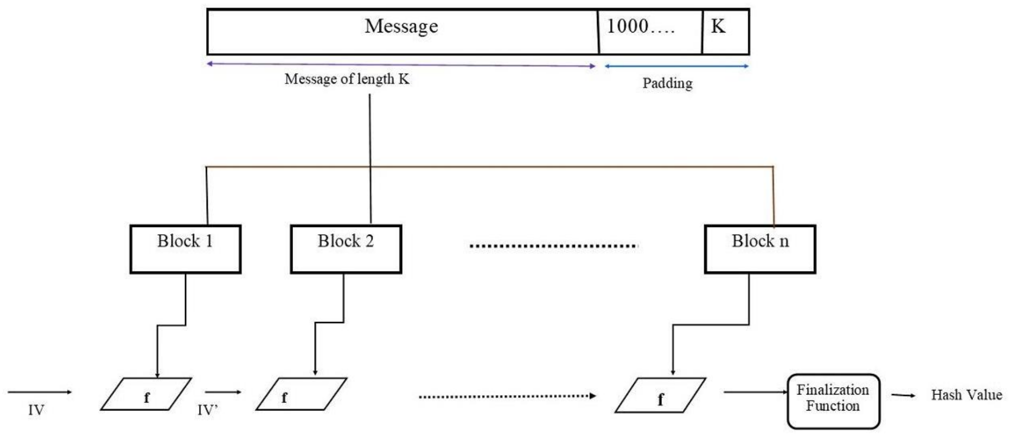
9) We can treat M = message, K = secret. When we apply SHA256 to K || M, it will hash M' = M || Padding. This is because SHA256 goes through blocks iteratively. Here, the padding bits will be filled into the last block by the hash function. So, with no padding involved, SHA256(K || M) will hash M' = M. Now, M' = M || X. So, it generates a valid signature without knowing a secret key. So, if we know SHA256(secret || constant), we can then compute SHA256 as SHA256(secret || constant || newData).

10) Yes, we would need to know the length of the key to execute an attack, the information/content of the message is not needed. The attacker would only require two attributes to initiate the attack, the hash message, and the length of the message.

11) The padding used will be when it multiplies the hash with the exact length to form the encoded message. The hash value for N with the given string "extra content" is "54d7d4c9f8c16de2dd189eadeeb239359f253fc59221665b7e22b6e6e23a3ce3". Therefore, the hash(K:M:N) is: abcd9313x:1234567890123456789012345678901234567890:54d7d4c9f8c16de2dd189eadeeb239359f253fc59221665b7e22b6e6e23a3ce3.

12) Alice would be able to compute HMAC (K, “This is Alice”) and send the result to Bob. Bob will then be able to verify the result if it is correct by computing the same, HMAC (K, “This is Alice”) by himself. Alice could also send the secret number K to Bob, and he can decrypt it to verify it is Alice, and if she knows the secret number. Alice could also request Bob to compute HMAC (K, “Alice’s message”) himself and she could send her results to Bob to verify each other. Alternatively, Alice could send Bob a challenge whereby Bob would have to compute a random string challenge through HMAC (K, challenge) and he would then send the results back to Alice for verification.

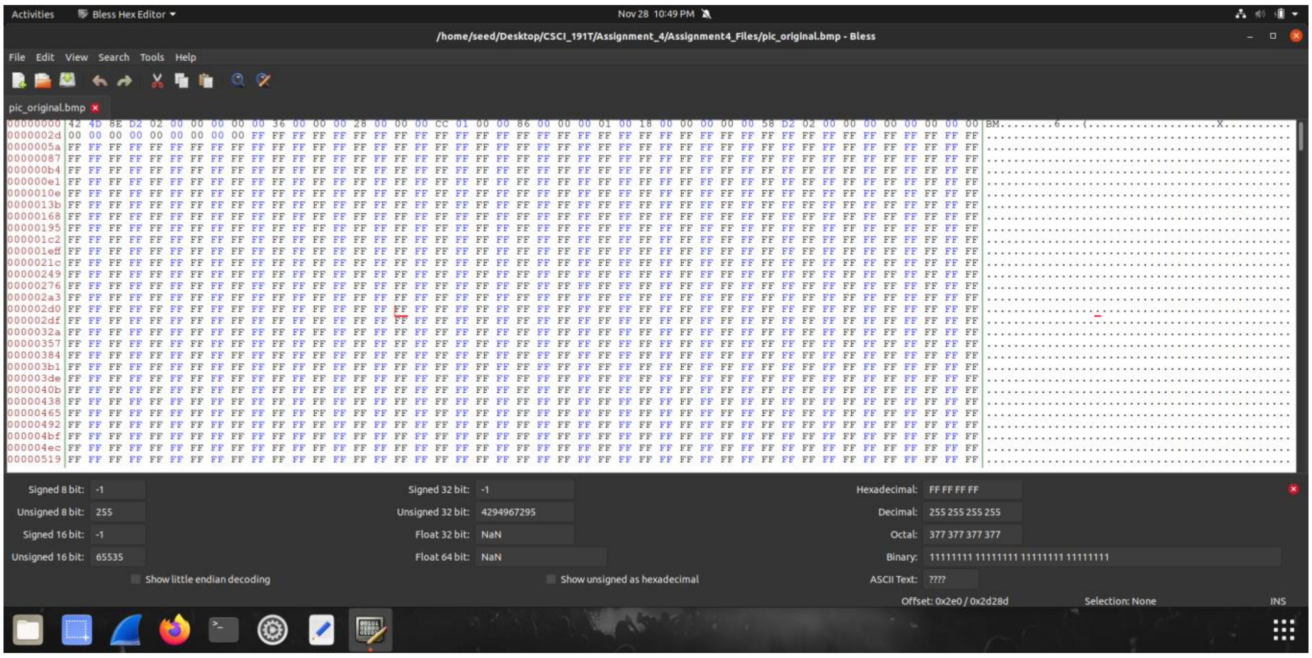
13) The Merkle-Damgard construction is a method to create a cryptographic hash function that uses a one-way compression function. The construction is grounded on the notion that if the compression function is collision-resistant, so will the hash function too. The construction has been used to design various hash functions like MD5, SHA-1, and SHA-2. The process of the construction begins with the lengthening of the input message to a multiple of some fixed number of bits. Since the compression function only works on fixed-length inputs. From the diagram below, we get two blocks in one of the Merkle-Damgard system. The first is a padding block with 1 and 0, the second is the message size block. The input message is split into blocks of a set length. After that, the blocks are processed with a one-way compression function (f) and an initial vector (IV). The initial vector is a predefined value that changes with every iteration, and finally produces the hash. This will then be processed further with the finalization function, which ensures a fixed-length output.



14) If Bob found two different messages M1 and M2 where they both have the same SHA256 hash, it is extremely rare to do so. This is because SHA256 is made to create unique strings of fixed length 256 bit whenever an input of any size is passed into it. Hashing algorithms are considered secure unless it intentionally generates collisions. In the case for the pair M3 and M4, it is not possible to find another pair where SHA256 hashes resulted in being the same.

Section 2:

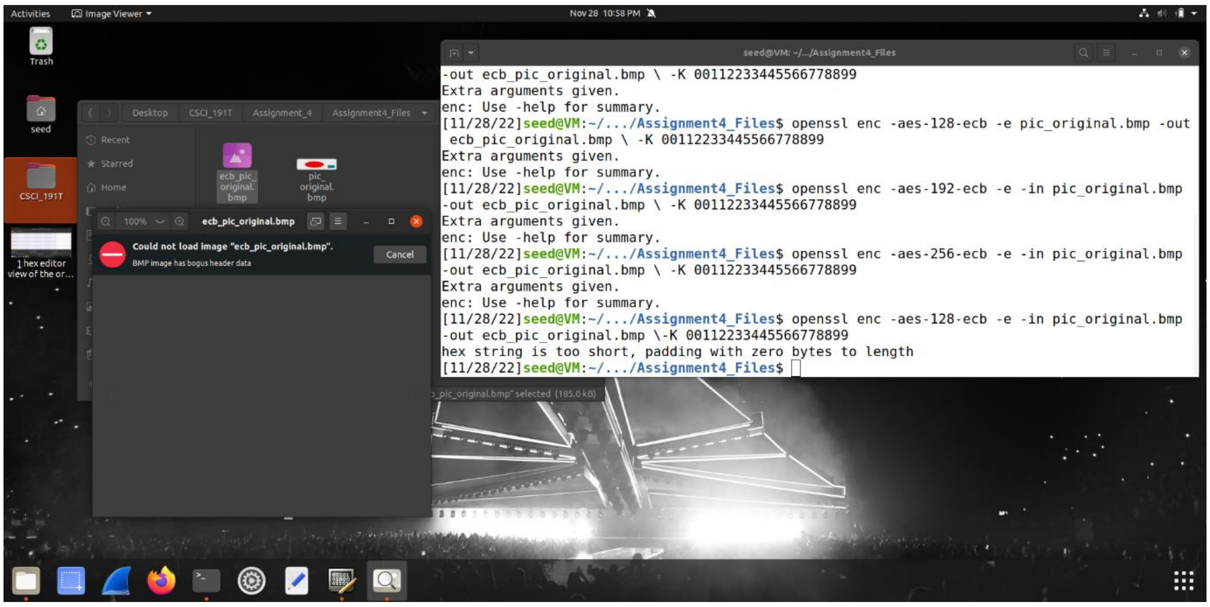
Part 1 (Using pic_original.bmp):


Activities
Bless Hex Editor
Nov 28 10:49 PM
/home/seed/Desktop/CSCI_191T/Assignment_4/Assignment4_Files/pic_original.bmp - Bless
File
Edit
View
Search
Tools
Help
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 00 18 00 00 00 00 00 58 D2 02 00 00 00 00 00 00 00 00 BM........6...(...................X..........
0000002d 00 00 00 00 00 00 00 00 00 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 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 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 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 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 .............................................
000000e1 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 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
0000013b 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 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 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 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 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 .............................................
000001ef 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 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
00000249 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
00000276 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 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
000002d0 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
000002fd 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 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
00000357 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
00000384 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
000003b1 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
000003de 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
0000040b 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
00000438 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
00000465 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
00000492 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
000004bf 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
000004ec 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
00000519 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 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .............................................
```

```
Signed 8 bit: -1            Signed 32 bit: -1                Hexadecimal: FF FF FF FF
Unsigned 8 bit: 255         Unsigned 32 bit: 4294967295      Decimal: 255 255 255 255
Signed 16 bit: -1           Float 32 bit: NaN                Octal: 377 377 377 377
Unsigned 16 bit: 65535      Float 64 bit: NaN                Binary: 11111111 11111111 11111111 11111111
    Show little endian decoding    Show unsigned as hexadecimal    ASCII Text: ????
                                                 Offset: 0x2e0 / 0x2d28d    Selection: None    INS
```

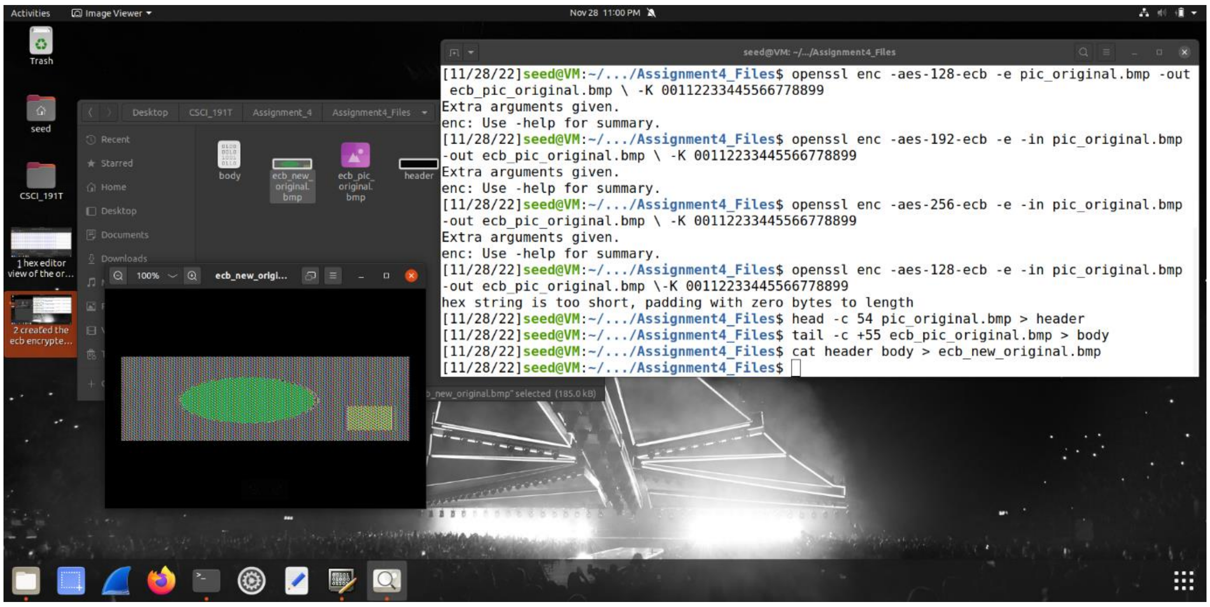
Step 1: Observing the hex editor of the original image before any encryption occurs.



```
Could not load image "ecb_pic_original.bmp".
BMP image has bogus header data
```

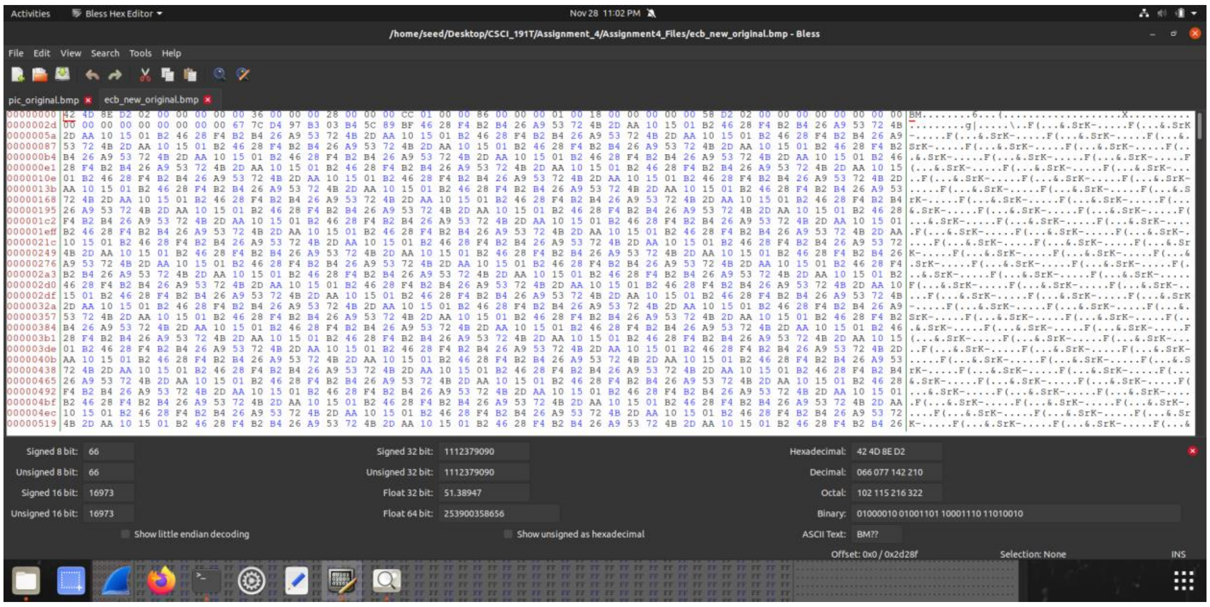
```
-out ecb_pic_original.bmp \ -K 00112233445566778899
Extra arguments given.
enc: Use -help for summary.
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-128-ecb -e pic_original.bmp -out
 ecb_pic_original.bmp \ -K 00112233445566778899
Extra arguments given.
enc: Use -help for summary.
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-192-ecb -e -in pic_original.bmp
-out ecb_pic_original.bmp \ -K 00112233445566778899
Extra arguments given.
enc: Use -help for summary.
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-256-ecb -e -in pic_original.bmp
-out ecb_pic_original.bmp \ -K 00112233445566778899
Extra arguments given.
enc: Use -help for summary.
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-128-ecb -e -in pic_original.bmp
-out ecb_pic_original.bmp \-K 00112233445566778899
hex string is too short, padding with zero bytes to length
[11/28/22]seed@VM:~/.../Assignment4_Files$ 
```

Step 2: After creating the ecb encryption of the original image, the image cannot be opened on the image viewer.



```
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-128-ecb -e pic_original.bmp -out
 ecb_pic_original.bmp \ -K 00112233445566778899
Extra arguments given.
enc: Use -help for summary.
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-192-ecb -e -in pic_original.bmp
-out ecb_pic_original.bmp \ -K 00112233445566778899
Extra arguments given.
enc: Use -help for summary.
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-256-ecb -e -in pic_original.bmp
-out ecb_pic_original.bmp \ -K 00112233445566778899
Extra arguments given.
enc: Use -help for summary.
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-128-ecb -e -in pic_original.bmp
-out ecb_pic_original.bmp \-K 00112233445566778899
hex string is too short, padding with zero bytes to length
[11/28/22]seed@VM:~/.../Assignment4_Files$ head -c 54 pic_original.bmp > header
[11/28/22]seed@VM:~/.../Assignment4_Files$ tail -c +55 ecb_pic_original.bmp > body
[11/28/22]seed@VM:~/.../Assignment4_Files$ cat header body > ecb_new_original.bmp
[11/28/22]seed@VM:~/.../Assignment4_Files$ 
```

Step 3: After combining the header of the original image and the body of the encrypted image, we can get an encrypted version of the image to show in the image viewer that is slightly encrypted and can still be observed as the shapes are still visible but are not in the original color as before.

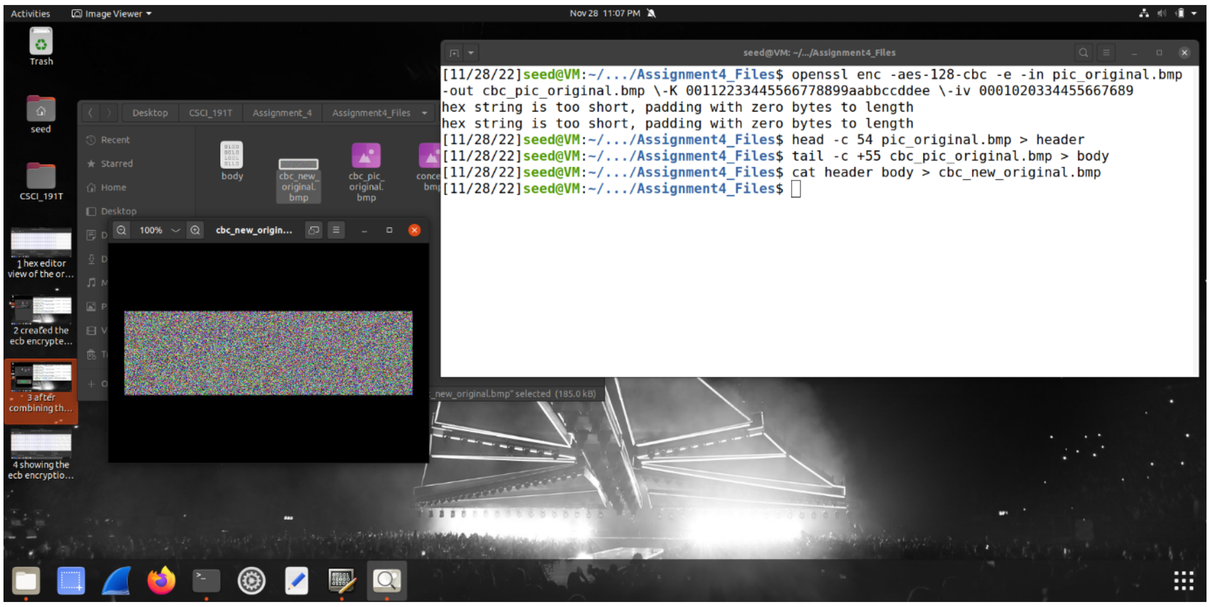


/home/seed/Desktop/CSCI_191T/Assignment_4/Assignment4_Files/ecb_new_original.bmp - Bless

```
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 00 18 00 00 00 00 00 58 D2 02 00 00 00 00 00 00 00 00 BM........6...(...................X..........
0000002d 00 00 00 00 00 00 00 00 00 67 7C D4 97 B3 03 B4 5C 89 BF 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B .........g|.....\..F(...&.SrK-.....F(...&.SrK
0000005a 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 -.....F(...&.SrK-.....F(...&.SrK-.....F(...&.
00000087 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 SrK-.....F(...&.SrK-.....F(...&.SrK-.....F(..
000000b4 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 .&.SrK-.....F(...&.SrK-.....F(...&.SrK-.....F
000000e1 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 (...&.SrK-.....F(...&.SrK-.....F(...&.SrK-...
0000010e 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D ..F(...&.SrK-.....F(...&.SrK-.....F(...&.SrK-
0000013b AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 .....F(...&.SrK-.....F(...&.SrK-.....F(...&.S
00000168 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 rK-.....F(...&.SrK-.....F(...&.SrK-.....F(...
00000195 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 &.SrK-.....F(...&.SrK-.....F(...&.SrK-.....F(
000001c2 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 ...&.SrK-.....F(...&.SrK-.....F(...&.SrK-....
000001ef B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA .F(...&.SrK-.....F(...&.SrK-.....F(...&.SrK-.
0000021c 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 ....F(...&.SrK-.....F(...&.SrK-.....F(...&.Sr
00000249 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 K-.....F(...&.SrK-.....F(...&.SrK-.....F(...&
00000276 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 .SrK-.....F(...&.SrK-.....F(...&.SrK-.....F(.
000002a3 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 ..&.SrK-.....F(...&.SrK-.....F(...&.SrK-.....
000002d0 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 F(...&.SrK-.....F(...&.SrK-.....F(...&.SrK-..
000002df 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B ...F(...&.SrK-.....F(...&.SrK-.....F(...&.SrK
0000032a 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 -.....F(...&.SrK-.....F(...&.SrK-.....F(...&.
00000357 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 SrK-.....F(...&.SrK-.....F(...&.SrK-.....F(..
00000384 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 .&.SrK-.....F(...&.SrK-.....F(...&.SrK-.....F
000003b1 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 (...&.SrK-.....F(...&.SrK-.....F(...&.SrK-...
000003de 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D ..F(...&.SrK-.....F(...&.SrK-.....F(...&.SrK-
0000040b AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 .....F(...&.SrK-.....F(...&.SrK-.....F(...&.S
00000438 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 rK-.....F(...&.SrK-.....F(...&.SrK-.....F(...
00000465 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 &.SrK-.....F(...&.SrK-.....F(...&.SrK-.....F(
00000492 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 ...&.SrK-.....F(...&.SrK-.....F(...&.SrK-....
000004bf B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA .F(...&.SrK-.....F(...&.SrK-.....F(...&.SrK-.
000004ec 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 ....F(...&.SrK-.....F(...&.SrK-.....F(...&.Sr
00000519 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 A9 53 72 4B 2D AA 10 15 01 B2 46 28 F4 B2 B4 26 K-.....F(...&.SrK-.....F(...&.SrK-.....F(...&
```

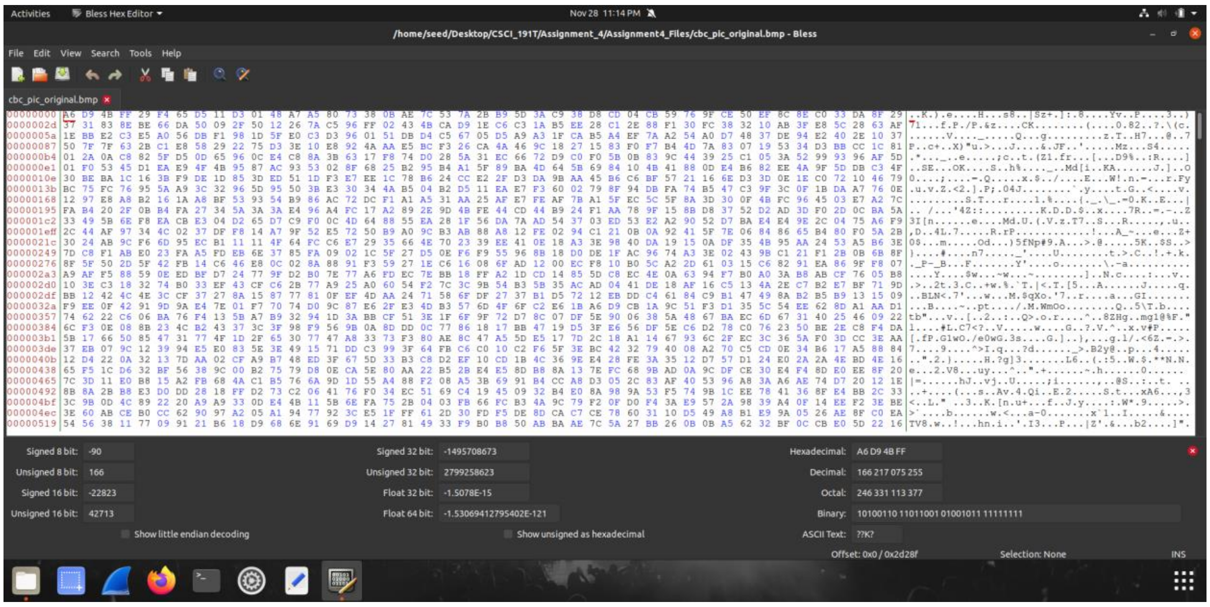
```
Signed 8 bit: 66              Signed 32 bit: 1112379090      Hexadecimal: 42 4D 8E D2
Unsigned 8 bit: 66            Unsigned 32 bit: 1112379090    Decimal: 066 077 142 210
Signed 16 bit: 16973          Float 32 bit: 51.38947         Octal: 102 115 216 322
Unsigned 16 bit: 16973        Float 64 bit: 253900358656     Binary: 01000010 01001101 10001110 11010010
Show little endian decoding   Show unsigned as hexadecimal   ASCII Text: BM??
Offset: 0x0 / 0x2d28f         Selection: None                INS
```

Step 4: This was the editor showing the ECB encryption that has taken place for the specific file.



```
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-128-cbc -e -in pic_original.bmp
-out cbc_pic_original.bmp \-K 00112233445566778899aabbccddee \-iv 0001020334455667689
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[11/28/22]seed@VM:~/.../Assignment4_Files$ head -c 54 pic_original.bmp > header
[11/28/22]seed@VM:~/.../Assignment4_Files$ tail -c +55 cbc_pic_original.bmp > body
[11/28/22]seed@VM:~/.../Assignment4_Files$ cat header body > cbc_new_original.bmp
[11/28/22]seed@VM:~/.../Assignment4_Files$ 
```

Step 5: When doing the CBC encryption, we can see that the image in the image viewer is now completely hidden, we are unable to see the original image and it is all blurred out.

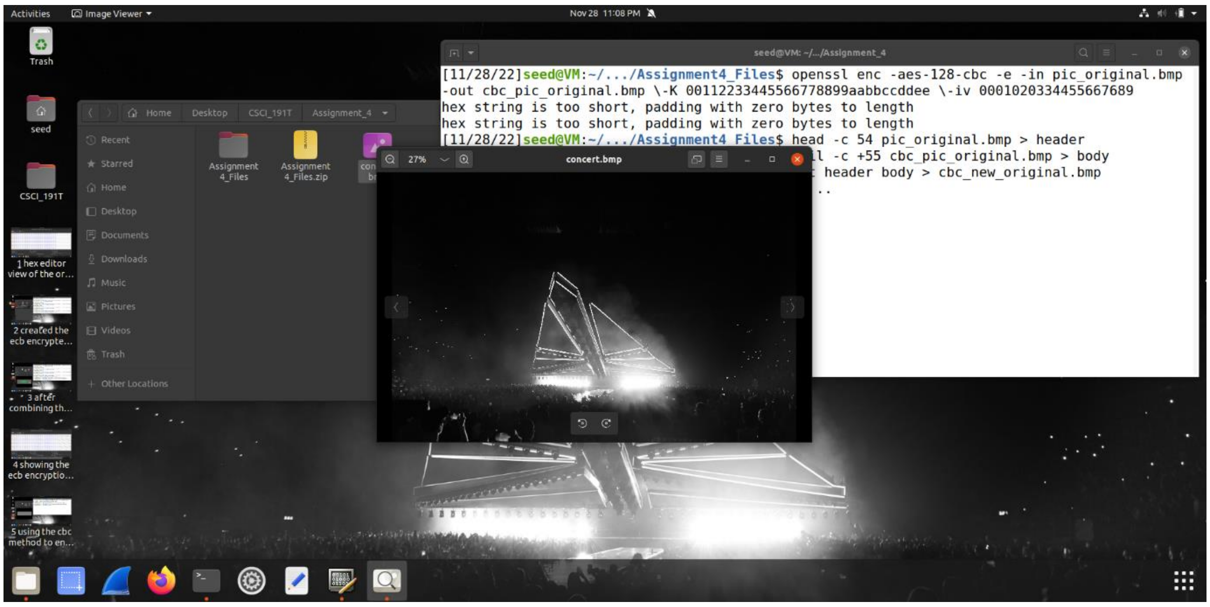

Activities
Bless Hex Editor
Nov 28 11:14 PM
/home/seed/Desktop/CSCI_191T/Assignment_4/Assignment4_Files/cbc_pic_original.bmp - Bless
File
Edit
View
Search
Tools
Help
cbc_pic_original.bmp

```
00000000 A6 D9 4B FF 29 F4 65 D5 11 D3 01 48 A7 A5 80 73 38 0B AE 7C 53 7A 2B B9 5D 3A C9 38 D8 CD 04 CB 59 76 9F CE 50 EF 8C 8E C0 33 DA 8F 29 ..K.).e....H...s8..|Sz+.]:.8....Yv..P....3..)
0000002d 37 31 83 8E BE 66 DA 50 09 2F 50 12 26 7A C5 96 FF 02 43 4B CA D9 1E C6 C3 1A B5 EE 28 C1 2E 88 F1 30 FC 38 32 10 AB 3F E8 5C 28 63 AF 71...f.P./P.&z....CK........(....0.82..?.\(c.
0000005a 1E BB E2 C3 E5 A0 56 DB F1 98 1D 5F E0 C3 D3 96 01 51 DB D4 C5 67 05 D5 A9 A3 1F CA B5 A4 EF 7A A2 54 A0 D7 48 37 DE 94 E2 40 2E 10 37 ......V...._.....Q...g.........z.T..H7...@..7
00000087 50 7F 7F 63 2B C1 E8 58 29 22 75 D3 3E 10 E8 92 4A AA E5 BC F3 26 CA 4A 46 9C 18 27 15 83 F0 F7 B4 4D 7A 83 07 19 53 34 D3 BB CC 1C 81 P..c+..X)"u.>...J....&.JF..'.....Mz...S4.....
000000b4 01 2A 0A C8 82 5F D5 0D 65 96 0C E4 C8 8A 3B 63 17 F8 74 D0 28 5A 31 EC 66 72 D9 C0 F0 5B 0B 83 9C 44 39 25 C1 05 3A 52 99 93 96 AF 5D .*..._..e.....;c..t.(Z1.fr...[...D9%..:R....]
000000e1 01 F0 53 45 D1 EA E9 4F 4B 95 87 AC 93 53 02 8F 68 25 B2 95 B4 A1 5F 89 BA 4D 64 5B 69 84 10 4B 41 88 0D E4 B6 82 EE 4A 9F 5D DB C3 4F ..SE...OK....S..h%...._..Md[i..KA......J.]..O
0000010e 30 BE BA 1C 16 3B F9 DE 1D 85 3D ED 51 1D F3 E7 EE 1C 78 B6 24 CC E2 2F D3 DA 9B AA 45 B6 C6 BF 57 21 16 6E D3 3D 0E 1E C0 72 10 46 79 0....;....=.Q.....x.$../....E...W!.n.=....r.Fy
0000013b BC 75 FC 76 95 5A A9 3C 32 96 5D 95 50 3B E3 30 34 4A B5 04 B2 D5 11 EA E7 F3 60 02 79 8F 94 DB FA 74 B5 47 C3 9F 3C 0F 1B DA A7 76 0E .u.v.Z.<2.].P;.04J.......`.y....t.G..<....v.
00000168 12 97 E8 A8 B2 16 1A A8 BF 53 93 54 B9 86 AC 72 DC F1 A1 A5 31 AA 25 AF E7 FE AF 7B A1 5F EC 5C 5F 8A 3D 30 0F 4B FC 96 45 03 E7 A2 7C .........S.T...r....1.%....{._.\_.=0.K..E...|
00000195 FA B4 20 2F 0B B4 FA 27 34 5A 3A 3A E4 96 A4 FC 17 A2 89 2E 9D 4B FE 44 CD 44 B9 24 F1 AA 78 9F 15 8B D8 37 52 D2 AD 3D F0 2D 0C BA 5A .. /...'4Z::.........K.D.D.$..x....7R..=.-..Z
000001c2 33 49 5B 6E F8 EA CB E3 04 D2 65 D7 C9 F0 0C 4D 64 88 55 EA 28 1F 56 DA 7A AD 54 37 03 ED 53 E2 A2 90 52 D7 BA E4 E4 9E 2C 04 75 A6 F9 3I[n......e....Md.U.(.V.z.T7..S...R.....,.u..
000001ef 2C 44 AF 97 34 4C 02 37 DF F8 14 A7 9F 52 E5 72 50 B9 A0 9C B3 AB 88 A8 12 FE 02 94 C1 21 0B 0A 92 41 5F 7E 06 84 86 65 B4 80 F0 5A 2B ,D..4L.7.....R.rP............!...A_~...e...Z+
0000021c 30 24 AB 9C F6 6D 95 EC B1 11 11 4F 64 FC C6 E7 29 35 66 4E 70 23 39 EE 41 0E 18 A3 3E 98 40 DA 19 15 0A DF 35 4B 95 AA 24 53 A5 B6 3E 0$...m.....Od...)5fNp#9.A...>.@.....5K..$S..>
00000249 7D C8 F1 AB E0 23 FA A5 FD EB 6E 37 85 FA 09 02 1C 5F 27 D5 0E F6 F9 55 96 8B 18 D0 DE 1F AC 96 74 A3 3E 02 43 9B C1 21 F1 2B 0B 6B 8F }....#....n7....._'....U........t.>.C..!.+.k.
00000276 8F 5F 50 2D 5F 42 FB 14 C6 46 E8 0C 02 8A 88 91 F3 59 27 1E C6 16 08 6F AD 12 00 EC F8 10 B0 5C A2 2D 61 03 15 C6 82 91 EA 86 9F F8 07 ._P-_B...F.......Y'....o.......\.-a.........
000002a3 A9 AF F5 88 59 0E ED BF D7 24 77 9F D2 B0 7E 77 A6 FD EC 7E BB 18 FF A2 1D CD 14 85 5D C8 EC 4E 0A 63 94 F7 B0 A0 3A B8 AB CF 76 05 B8 ....Y....$w...~w...~.........]..N.c....:...v..
000002d0 10 3E C3 18 32 74 B0 33 EF 43 CF C6 2B 77 A9 25 A0 60 54 F2 7C 3C 9B 54 B3 5B 35 AC AD 04 41 DE 18 AF 16 C5 13 4A 2E C7 B2 E7 BF 71 9D .>..2t.3.C..+w.%.`T.|<.T.[5...A......J.....q.
000002fd BB 12 42 4C 4E 3C CF 37 27 8A 15 87 77 81 0F EF 4D AA 24 71 58 6F DF 27 37 B1 D5 72 12 EB DD C4 61 84 C9 B1 47 49 8A B2 B5 B9 13 15 09 ..BLN<.7'...w...M.$qXo.'7..r....a...GI......
0000032a F9 EE 0F 42 91 9D 9A E4 7E 01 F7 70 74 D0 9C 87 E6 2F E3 4D B3 57 6D 4F 6F C2 E6 1B A6 D9 CB 1A 9C 51 F3 D1 35 5C 54 EE 62 8D A1 AA D1 ...B....~..pt..../.M.WmOo.......Q..5\T.b....
00000357 74 62 22 C6 06 BA 76 F4 13 5B A7 B9 32 94 1D 3A BB CF 51 3E 1F 6F 9F 72 D7 8C 07 DF 5E 90 06 38 5A 48 67 BA EC 6D 67 31 40 25 46 09 22 tb"...v..[..2..:..Q>.o.r....^..8ZHg..mg1@%F."
00000384 6C F3 0E 08 8B 23 4C B2 43 37 3C 3F 98 F9 56 9B 0A 8D DD 0C 77 86 18 17 BB 47 19 D5 3F E6 56 DF 5E C6 D2 78 C0 76 23 50 BE 2E C8 F4 DA l....#L.C7<?..V.....w....G..?.V.^..x.v#P.....
000003b1 5B 17 66 50 85 47 31 77 4F 1D 2F 65 30 77 47 A8 33 73 F3 80 AE 8C 47 A5 5D E5 17 7D 2C 18 A1 14 67 93 6C 2F EC 3C 36 5A F0 3D CC 3E AA [.fP.G1wO./e0wG.3s....G.]..},...g.l/.<6Z.=.>.
000003de 37 EB 07 9C 12 39 94 E5 E0 83 5E 3E 49 15 71 DD C3 99 3F 64 FB C6 C0 10 C2 F6 5F 3E BC 42 32 79 40 08 A2 70 C5 CD 0E 34 B6 17 A5 88 84 7....9....^>I.q...?d......_>.B2y@..p...4.....
0000040b 12 D4 22 0A 32 13 7D AA 02 CF A9 B7 48 ED 3F 67 5D 33 B3 C8 D2 EF 10 CD 1B 4C 36 9E E4 28 FE 3A 35 12 D7 57 D1 24 E0 2A 2A 4E BD 4E 16 .."..2.}.....H.?g]3.......L6..(.:5..W.$.**N.N.
00000438 65 F5 1C D6 32 BF 56 38 9C 00 B2 75 79 D8 0E CA 5E 80 AA 22 B5 2B E4 E5 8D B8 8A 13 7E FC 68 9B AD 0A 9C DF CE 30 E4 F4 8D E0 EE 8F 20 e...2.V8...uy...^..".+......~.h......0......
00000465 7C 3D 11 E0 B8 15 A2 FB 68 4A C1 B5 76 6A 9D 1D 55 A4 88 F2 08 A5 3B 69 91 B4 CC A8 D3 05 2C 83 AF 40 53 96 A8 3A A6 AE 74 D7 20 12 1E |=......hJ..vj..U.....;i.......,..@S..:..t. ..
00000492 8B 8A 2B B8 E3 D0 DD 28 18 FF D2 73 C2 06 41 76 F0 34 EC 51 69 C4 19 45 09 32 B4 E0 8A 98 9A 53 F5 74 9B 1C EE 78 41 36 8F E4 BB 2C 33 ..+....(...s..Av.4.Qi..E.2.....S.t...xA6...,3
000004bf 3C 9B 0D 4C 89 22 20 A9 A9 33 0D E4 4B 11 5B 6E FA 75 2B 04 03 FB 66 FC B3 4A 9C 79 F2 0F D0 F4 3A E9 57 2A 98 39 A4 0F 14 EE F2 3E BE <..L." ..3..K.[n.u+...f..J.y....:.W*.9.....>.
000004ec 3E 60 AB CE B0 CC 62 90 97 A2 05 A1 94 77 92 3C E5 1F FF 61 2D 30 FD F5 DE 8D CA C7 CE 78 60 31 10 D5 49 A8 B1 E9 9A 05 26 AE 8F C0 EA >`....b......w.<...a-0.......x`1..I.....&....
00000519 54 56 38 11 77 09 91 21 B6 18 D9 68 6E 91 69 D9 14 27 81 49 33 F9 B0 B8 50 AB BA AE 7C 5A 27 BB 26 0B 0B A5 62 32 BF 0C CB E0 5D 22 16 TV8.w..!...hn.i..'.I3...P...|Z'.&...b2....]".
```

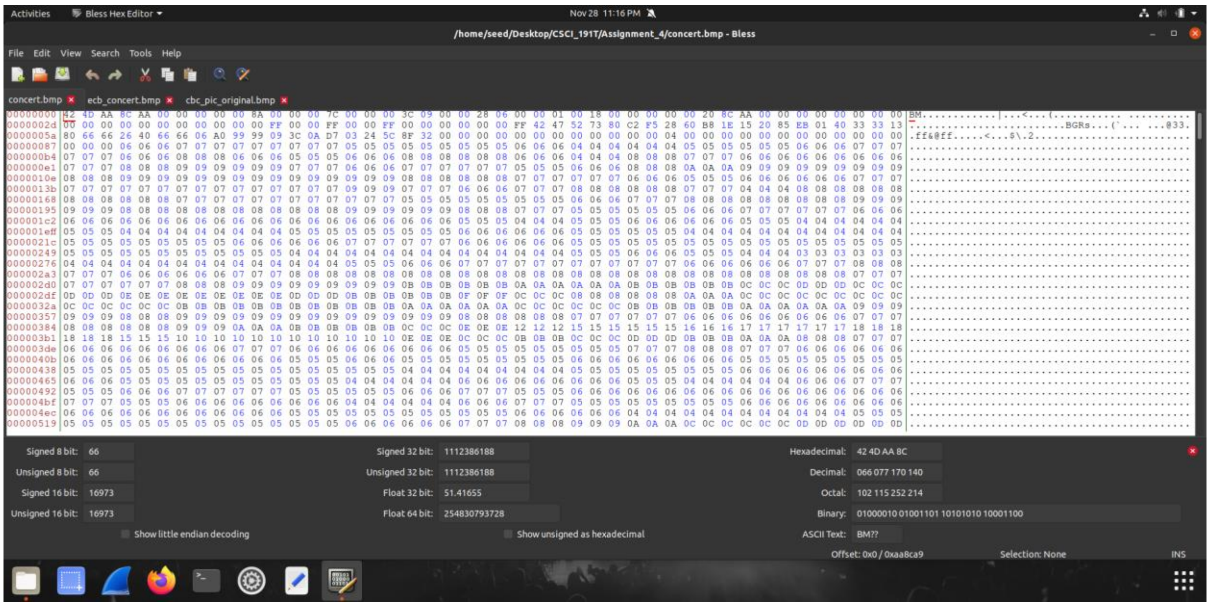
```
Signed 8 bit: -90                 Signed 32 bit: -1495708673                 Hexadecimal: A6 D9 4B FF
Unsigned 8 bit: 166               Unsigned 32 bit: 2799258623                Decimal: 166 217 075 255
Signed 16 bit: -22823             Float 32 bit: -1.5078E-15                  Octal: 246 331 113 377
Unsigned 16 bit: 42713            Float 64 bit: -1.53069412795402E-121       Binary: 10100110 11011001 01001011 11111111
   Show little endian decoding       Show unsigned as hexadecimal            ASCII Text: ??K?
                                                                             Offset: 0x0 / 0x2d28f     Selection: None     INS
```

Step 6: In the CBC mode, once again the image's hex editor displays different variables.

Part 2 (Using own image):



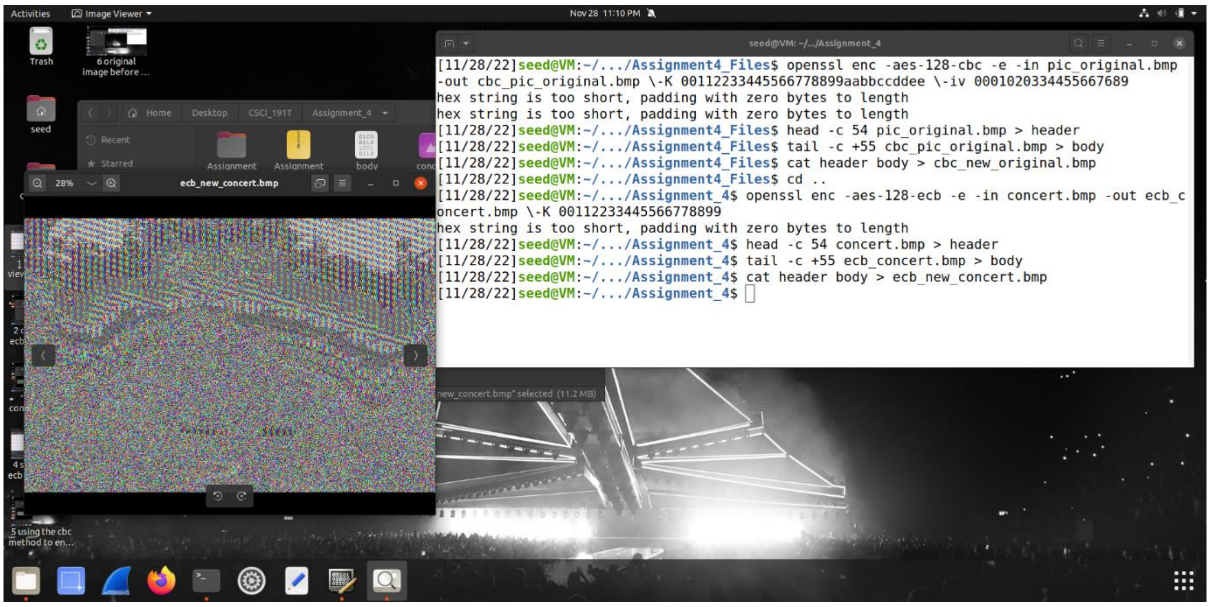
```
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-128-cbc -e -in pic_original.bmp
-out cbc_pic_original.bmp \-K 00112233445566778899aabbccddee \-iv 0001020334455667689
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[11/28/22]seed@VM:~/.../Assignment4_Files$ head -c 54 pic_original.bmp > header
                                         il -c +55 cbc_pic_original.bmp > body
                                          header body > cbc_new_original.bmp
                                         ..
```



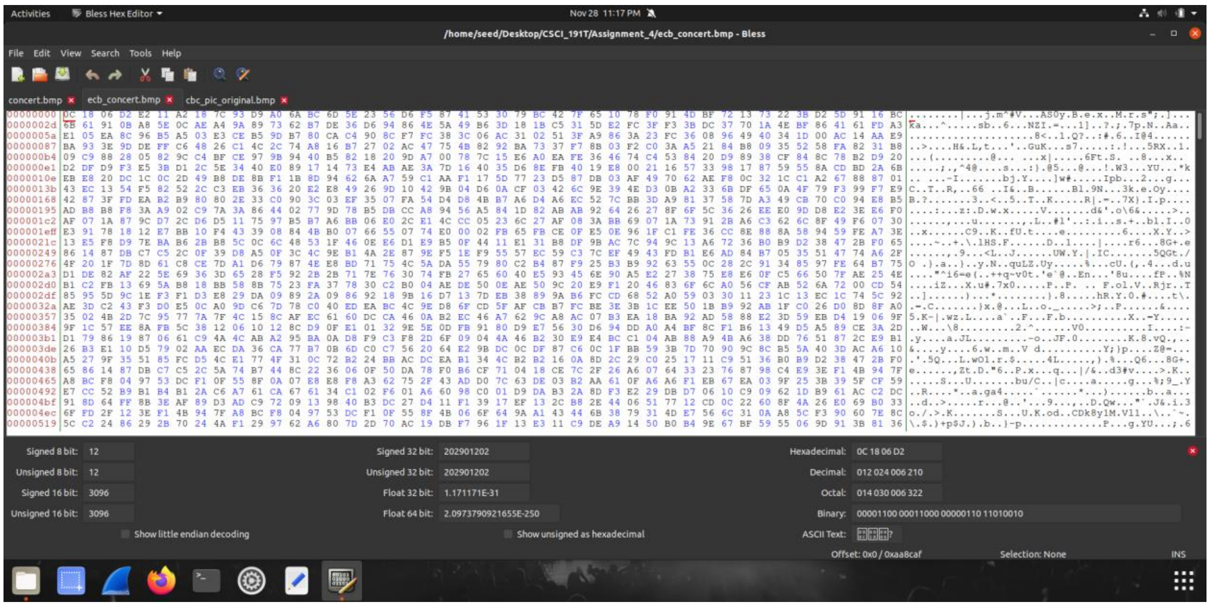
```
00000000 42 4D AA 8C AA 00 00 00 00 00 8A 00 00 00 7C 00 00 00 3C 09 00 00 28 06 00 00 01 00 18 00 00 00 00 00 20 8C AA 00 00 00 00 00 00 00 00 BM............|...<...(......................
0000002d 00 00 00 00 00 00 00 00 00 00 00 FF 00 00 FF 00 00 FF 00 00 00 00 00 00 FF 42 47 52 73 80 C2 F5 28 60 B8 1E 15 20 85 EB 01 40 33 33 13 .........................BGRs...(`... ...@33.
0000005a 80 66 66 26 40 66 66 06 A0 99 99 09 3C 0A D7 03 24 5C 8F 32 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 .ff&@ff.....<...$\.2.........................
00000087 00 00 00 06 06 06 07 07 07 07 07 07 07 07 07 05 05 05 05 05 05 05 05 05 06 06 06 04 04 04 04 04 04 05 05 05 05 05 05 06 06 06 07 07 07 .............................................
000000b4 07 07 07 06 06 06 08 08 08 06 06 06 05 05 05 06 06 06 08 08 08 08 08 08 06 06 06 04 04 04 08 08 08 07 07 07 06 06 06 06 06 06 06 06 06 .............................................
000000e1 07 07 07 08 08 08 09 09 09 09 09 09 07 07 07 06 06 06 07 07 07 07 07 07 05 05 05 06 06 06 08 08 08 0A 0A 0A 09 09 09 09 09 09 09 09 09 .............................................
0000010e 08 08 08 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 08 08 08 08 08 08 07 07 07 07 07 07 06 06 06 05 05 05 06 06 06 06 06 06 07 07 07 .............................................
0000013b 07 07 07 07 07 07 07 07 07 07 07 07 07 07 07 09 09 09 07 07 07 06 06 06 07 07 07 08 08 08 08 08 08 07 07 07 04 04 04 08 08 08 08 08 08 .............................................
00000168 08 08 08 08 08 08 07 07 07 07 07 07 07 07 07 07 07 07 05 05 05 05 05 05 05 05 05 06 06 06 07 07 07 08 08 08 08 08 08 08 08 08 09 09 09 .............................................
00000195 09 09 09 08 08 08 08 08 08 08 08 08 08 08 08 09 09 09 09 09 09 08 08 08 07 07 07 05 05 05 05 05 05 06 06 06 07 07 07 07 07 07 06 06 06 .............................................
000001c2 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 05 05 05 04 04 04 05 05 05 06 06 06 06 06 06 05 05 05 04 04 04 04 04 04 .............................................
000001ef 05 05 05 04 04 04 04 04 04 04 04 04 05 05 05 05 05 05 05 05 05 06 06 06 06 06 06 05 05 05 05 05 05 04 04 04 04 04 04 04 04 04 04 04 04 .............................................
0000021c 05 05 05 05 05 05 05 05 05 06 06 06 06 06 06 07 07 07 07 07 07 06 06 06 06 06 06 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 .............................................
00000249 05 05 05 05 05 05 05 05 05 05 05 05 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 05 05 05 06 06 06 05 05 05 04 04 04 03 03 03 03 03 03 .............................................
00000276 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 05 05 05 06 06 06 07 07 07 07 07 07 07 07 07 07 07 07 06 06 06 06 06 06 07 07 07 08 08 08 .............................................
000002a3 07 07 07 06 06 06 06 06 06 07 07 07 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 07 07 07 .............................................
000002d0 07 07 07 07 07 07 08 08 08 09 09 09 09 09 09 09 09 09 0B 0B 0B 0B 0B 0B 0A 0A 0A 0A 0A 0A 0B 0B 0B 0B 0B 0B 0C 0C 0C 0D 0D 0D 0C 0C 0C .............................................
000002fd 0D 0D 0D 0E 0E 0E 0E 0E 0E 0E 0E 0E 0D 0D 0D 0B 0B 0B 0B 0B 0B 0F 0F 0F 0C 0C 0C 08 08 08 08 08 08 0A 0A 0A 0C 0C 0C 0C 0C 0C 0C 0C 0C .............................................
0000032a 0C 0C 0C 0C 0C 0C 0B 0B 0B 0B 0B 0B 0B 0B 0B 0B 0B 0B 0A 0A 0A 0A 0A 0A 0C 0C 0C 0C 0C 0C 0B 0B 0B 0B 0B 0B 0A 0A 0A 0A 0A 0A 09 09 09 .............................................
00000357 09 09 09 08 08 08 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 08 08 08 08 08 08 07 07 07 07 07 07 06 06 06 06 06 06 06 06 06 07 07 07 .............................................
00000384 08 08 08 08 08 08 09 09 09 0A 0A 0A 0B 0B 0B 0B 0B 0B 0C 0C 0C 0E 0E 0E 12 12 12 15 15 15 15 15 15 16 16 16 17 17 17 17 17 17 18 18 18 .............................................
000003b1 18 18 18 15 15 15 10 10 10 10 10 10 10 10 10 10 10 10 0E 0E 0E 0C 0C 0C 0B 0B 0B 0C 0C 0C 0D 0D 0D 0B 0B 0B 0A 0A 0A 08 08 08 07 07 07 .............................................
000003de 06 06 06 06 06 06 06 06 06 07 07 07 06 06 06 06 06 06 06 06 06 05 05 05 05 05 05 05 05 05 07 07 07 08 08 08 07 07 07 06 06 06 06 06 06 .............................................
0000040b 06 06 06 06 06 06 06 06 06 06 06 06 05 05 05 06 06 06 05 05 05 05 05 05 05 05 05 06 06 06 06 06 06 06 06 06 05 05 05 05 05 05 05 05 05 .............................................
00000438 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 04 04 04 04 04 04 04 04 04 05 05 05 05 05 05 05 05 05 06 06 06 06 06 06 06 06 06 .............................................
00000465 06 06 06 05 05 05 05 05 05 05 05 05 05 05 05 04 04 04 04 04 04 06 06 06 06 06 06 06 06 06 05 05 05 04 04 04 04 04 04 06 06 06 07 07 07 .............................................
00000492 05 05 05 06 06 06 07 07 07 07 07 07 05 05 05 05 05 05 06 06 06 07 07 07 05 05 05 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 .............................................
000004bf 07 07 07 05 05 05 06 06 06 06 06 06 06 06 06 04 04 04 04 04 04 06 06 06 07 07 07 05 05 05 05 05 05 05 05 05 06 06 06 06 06 06 06 06 06 .............................................
000004ec 06 06 06 06 06 06 06 06 06 06 06 06 05 05 05 05 05 05 05 05 05 05 05 05 06 06 06 06 06 06 04 04 04 04 04 04 04 04 04 04 04 04 05 05 05 .............................................
00000519 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 06 06 06 06 06 06 07 07 07 08 08 08 09 09 09 0A 0A 0A 0C 0C 0C 0C 0C 0C 0D 0D 0D 0D 0D 0D .............................................

Signed 8 bit: 66            Signed 32 bit: 1112386188        Hexadecimal: 42 4D AA 8C
Unsigned 8 bit: 66          Unsigned 32 bit: 1112386188      Decimal: 066 077 170 140
Signed 16 bit: 16973        Float 32 bit: 51.41655           Octal: 102 115 252 214
Unsigned 16 bit: 16973      Float 64 bit: 254830793728       Binary: 01000010 01001101 10101010 10001100
Show little endian decoding   Show unsigned as hexadecimal   ASCII Text: BM??
                                                  Offset: 0x0 / 0xaa8ca9    Selection: None    INS
```

Step 1: This is the original image that will be encrypted in both the ECB and CBC method.

The second image is the hex editor for the original image.



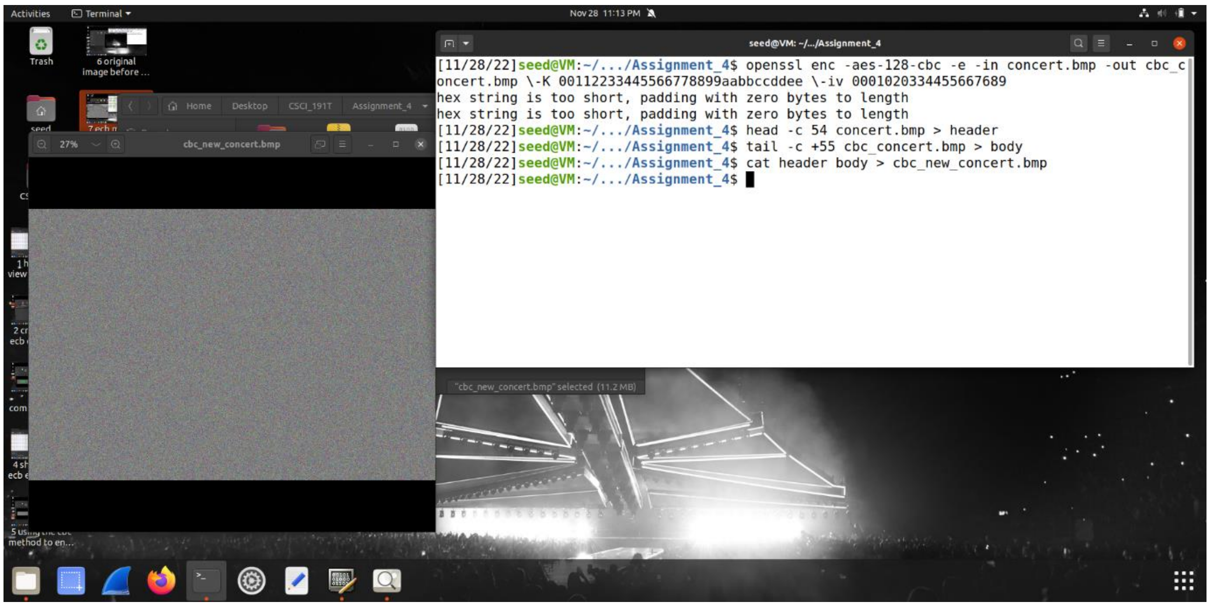
```
[11/28/22]seed@VM:~/.../Assignment4_Files$ openssl enc -aes-128-cbc -e -in pic_original.bmp
-out cbc_pic_original.bmp \-K 00112233445566778899aabbccddee \-iv 0001020334455667689
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[11/28/22]seed@VM:~/.../Assignment4_Files$ head -c 54 pic_original.bmp > header
[11/28/22]seed@VM:~/.../Assignment4_Files$ tail -c +55 cbc_pic_original.bmp > body
[11/28/22]seed@VM:~/.../Assignment4_Files$ cat header body > cbc_new_original.bmp
[11/28/22]seed@VM:~/.../Assignment4_Files$ cd ..
[11/28/22]seed@VM:~/.../Assignment_4$ openssl enc -aes-128-ecb -e -in concert.bmp -out ecb_c
oncert.bmp \-K 00112233445566778899
hex string is too short, padding with zero bytes to length
[11/28/22]seed@VM:~/.../Assignment_4$ head -c 54 concert.bmp > header
[11/28/22]seed@VM:~/.../Assignment_4$ tail -c +55 ecb_concert.bmp > body
[11/28/22]seed@VM:~/.../Assignment_4$ cat header body > ecb_new_concert.bmp
[11/28/22]seed@VM:~/.../Assignment_4$ 
```



```
00000000 0C 18 06 D2 E2 11 A2 18 7C 93 D9 A0 6A BC 6D 5E 23 56 D6 F5 87 41 53 30 79 BC 42 7F 65 10 78 F0 91 4D BF 72 13 73 22 3B D2 5D 91 16 BC ........|...j.m^#V...AS0y.B.e.x..M.r.s";.]...
0000002d 6B 61 91 0B A8 5E 0C AE A4 9A 89 73 62 B7 DE 36 D6 94 86 4E 5A 49 B6 3D 18 1B C5 31 5D E2 FC 3F F3 3B DC 37 70 1A 4E BF 86 41 61 FD A3 ka...^.....sb..6...NZI.=...1]..?.;.7p.N..Aa..
0000005a E1 05 EA 8C 96 B5 A5 03 E3 CE B5 9D B7 80 CA C4 90 8C F7 FC 38 3C 06 AC 31 02 51 3F A9 86 3A 23 FC 36 08 96 49 40 34 1D 00 AC 14 AA E9 ....................8<..1.Q?..:#.6..I@4......
00000087 BA 93 3E 9D DE FF C6 48 26 C1 4C 2C 74 A8 16 B7 27 02 AC 47 75 4B 82 92 BA 73 37 F7 8B 03 F2 C0 3A A5 21 84 B8 09 35 52 58 FA 82 31 B8 ..>....H&.L,t...'..GuK...s7.....:.!...5RX..1.
000000b4 09 C9 88 28 05 82 9C C4 BF CE 97 9B 94 40 B5 82 18 20 9D A7 00 78 7C 15 E6 A0 EA FE 36 46 74 C4 53 84 20 D9 89 38 CF 84 8C 78 B2 D9 20 ...(.........@... ...x|.....6Ft.S. ..8...x..
000000e1 D2 DF D9 F3 E5 3B D1 2C 5E 34 40 E0 89 17 14 73 E4 AB AE 3A 7D 16 40 35 D6 8E FB 40 19 E8 00 21 16 57 33 98 17 87 59 55 8A CD BD 2A 6B .....;.,^4@....s...:}.@5...@...!.W3...YU...*k
0000010e EB E8 20 DC 1C 0C 2D 49 B8 DE 8B F1 1B 8D 94 62 6A A7 59 C1 AA F1 17 5D 77 23 D5 87 DB 03 AF 49 70 62 AE F8 0C 32 1C C1 A2 67 88 87 01 .. ...-I.......bj.Y....]w#.....Ipb...2...g...
0000013b 43 EC 13 54 F5 82 52 2C C3 EB 36 36 20 E2 E8 49 26 9D 10 42 9B 04 D6 0A CF 03 42 6C 9E 39 4E D3 0B A2 33 6B DF 65 0A 4F 79 F3 99 F7 E9 C..T..R,..66 ..I&..B......Bl.9N...3k.e.Oy....
00000168 42 87 3F FD EA B2 B9 80 80 2E 33 C0 90 3C 03 EF 35 07 FA 54 D4 D8 4B B7 A6 D4 A6 EC 52 7C BB 3D A9 81 37 58 7D A3 49 CB 70 C0 94 E8 B5 B.?.......3..<..5..T..K.....R|.=..7X}.I.p....
00000195 AD B8 B8 F8 3A A9 02 C9 7A 3A 86 44 02 77 9D 78 B5 DB CC A8 94 56 A5 84 1D 82 AB AB 92 64 26 27 8F 6F 5C 36 26 EE E0 9D D8 E2 3E E6 F0 ....:...z:.D.w.x.....V.......d&'.o\6&.....>..
000001c2 AF 07 1A 87 9C D7 2C D6 D5 11 75 97 B5 B7 A6 BB 06 E0 2C E1 4C CC 05 23 6C 27 AF 08 3A BB 69 07 1A 73 91 2B A6 C3 62 6C 8F 49 F6 07 30 ......,...u.........,.L..#l'..:.i..s.+..bl.I..0
000001ef E3 91 78 18 12 E7 BB 10 F4 43 39 08 84 4B B0 07 66 55 07 74 E0 00 02 FB 65 FB CE 0F E5 0E 96 1F C1 FE 36 CC 8E 88 8A 58 94 59 FE A7 3E ..x......C9..K..fU.t....e..........6....X.Y..>
0000021c 13 E5 F8 D9 7E BA B6 2B B8 5C 0C 6C 48 53 1F 46 0E E6 D1 E9 B5 0F 44 11 E1 31 B8 DF 9B AC 7C 94 9C 13 A6 72 36 B0 B9 D2 38 47 2B F0 65 ....~..+.\.lHS.F......D..1....|....r6...8G+.e
00000249 86 14 87 DB C7 C5 2C 0F 39 D8 A5 0F 3C 4C 9E B1 4A 2E 87 9E F5 1E F9 55 57 EC 59 C3 7C EF 49 43 FD B1 E6 AD 84 B7 05 35 51 47 74 A6 2F ......,.9...<L..J......UW.Y.|.IC........5QGt./
00000276 4F 20 1F 7D 8D 61 C8 CE 7D A1 D6 79 87 4E E8 BD 71 75 4C 5A DA 55 79 80 C2 B4 87 F9 25 B3 B9 92 63 55 0C 28 2C 91 34 85 97 FE 64 B7 75 O .}.a..}..y.N..quLZ.Uy.....%...cU.(,.4...d.u
000002a3 D1 DE 82 AF 22 5E 69 36 3D 65 28 F5 92 2B 2B 71 7E 76 30 74 FB 27 65 60 40 E5 93 45 6E 90 A5 E2 27 38 75 E8 E6 0F C5 66 50 7F AE 25 4E ...."^i6=e(..++q~v0t.'e`@..En...'8u....fP..%N
000002d0 B1 C2 FB 13 69 5A B8 18 BB 58 8B 75 23 FA 37 78 30 C2 B0 04 AE DE 50 0E AE 50 9C 20 E9 F1 20 46 83 6F 6C A0 56 CF AB 52 6A 72 00 CD 54 ....iZ...X.u#.7x0.....P..P. .. F.ol.V..Rjr..T
000002fd 85 95 5D 9C 1E F3 F1 D3 E8 29 DA 09 89 2A 09 86 92 18 9B 16 D7 13 7D EB 38 89 9A B6 FC CD 68 52 A0 59 03 30 11 23 1C 13 EC 1C 74 5C 92 ..]......)...*.........}.8.....hR.Y.0.#....t\.
0000032a AE 3D C2 43 F3 D0 E5 0C A0 9D C6 7D 78 C0 40 ED EA BC 4C 9E D8 6F CD 5F AF CB B7 FC BE 3E 3B 1C EE 50 1B B9 92 AB 1F C0 26 D0 8D 8F A0 .=.C.......}x.@...L..o._......>;..P......&....
00000357 35 02 4B 2D 7C 95 77 7A 7F 4C 15 8C AF EC 61 60 DC CA 46 0A B2 EC 46 A7 62 9C A8 AC 07 B3 EA 18 BA 92 AD 58 88 E2 3D 59 EB D4 19 06 9F 5.K-|.wz.L....a`..F...F.b..........X..=Y.....
00000384 9F 1C 57 EE 8A FB 5C 38 12 06 10 12 8C D9 0F E1 01 32 9E 5E 0D FB 91 80 D9 E7 56 30 D6 94 DD A0 A4 BF 8C F1 B6 13 49 D5 A5 89 CE 3A 2D ..W...\8..........2.^......V0..........I....:-
000003b1 D1 79 86 19 87 06 61 C9 4A 4C AB A2 95 BA 0A D8 F9 C3 F8 2D 6F 09 04 4A 46 B2 30 E9 E4 BC C1 04 AB 88 A9 4B A6 38 DD 76 51 87 2C E9 B1 .y....a.JL.........-o..JF.0........K.8.vQ.,..
000003de 26 B3 E1 10 D5 79 02 AA EC DA 36 CA 77 B7 0B 6D C0 C7 56 20 64 E2 9B DC 0C DF 87 C6 0C 1F BB 59 3B 7D 70 90 9C 8C B5 5A 40 3D AC A6 10 &....y....6.w..m..V d..........Y;}p....Z@=...
0000040b A5 27 9F 35 51 85 FC D5 4C E1 77 4F 31 0C 72 B2 24 BB AC DC EA B1 34 4C B2 B2 16 0A 8D 2C 29 C0 25 17 11 C9 51 36 B0 B9 D2 38 47 2B F0 .'.5Q...L.wO1.r.$.....4L.....,).%...Q6...8G+.
00000438 65 86 14 87 DB C7 C5 2C 5A 74 B7 44 8C 22 36 06 0F 50 DA 78 F0 B6 CF 71 04 18 CE 7C 2F 26 A6 07 64 33 23 76 87 98 C4 E9 3E F1 4B 94 7F e.......,Zt.D."6..P.x...q...|/&..d3#v....>.K..
00000465 A8 BC F8 04 97 53 DC F1 0F 55 8F 0A 07 E8 E8 F8 A3 62 75 2F 43 AD D0 7C 63 DE 03 B2 AA 61 0F A6 A6 F1 EB 67 EA 03 9F 25 3B 39 5F CF 59 .....S...U.......bu/C..|c....a.....g...%;9_.Y
00000492 E7 CC 52 B9 B1 B4 B1 2A C6 A7 61 CA 67 61 34 C1 02 F6 01 A6 60 98 C0 01 D9 DA B3 2A 8D F3 E2 29 DB D7 06 10 C9 09 62 1D B9 61 AC C2 DC ..R....*..a.ga4.....`......*...).......b..a...
000004bf 91 8D 64 FF 8B 3E AF 89 D3 AD C9 72 09 13 98 40 B3 DC 27 D4 11 F1 39 17 EF 13 2C B8 2E 44 06 51 77 12 CD 0C 22 60 8F 4A 26 E0 69 B0 33 ..d..>.....r...@..'...9...,..D.Qw..."`.J&.i.3
000004ec 6F FD 2F 12 3E F1 4B 94 7F A8 BC F8 04 97 53 DC F1 0F 55 8F 4B 06 6F 64 9A A1 43 44 6B 38 79 31 4D E7 56 6C 31 0A A8 5C F3 90 60 7E 8C o./.>.K.......S...U.K.od..CDk8y1M.Vl1..\..`~.
00000519 5C C2 24 86 29 2B 70 24 4A F1 29 97 62 A6 80 7D 2D 70 AC 19 DB F7 96 1F 13 E3 11 C9 DE A9 14 50 B0 B4 9E 67 BF 59 55 06 9D 91 3B 81 36 \.$.)+p$J.).b..}-p.............P...g.YU...;.6
```

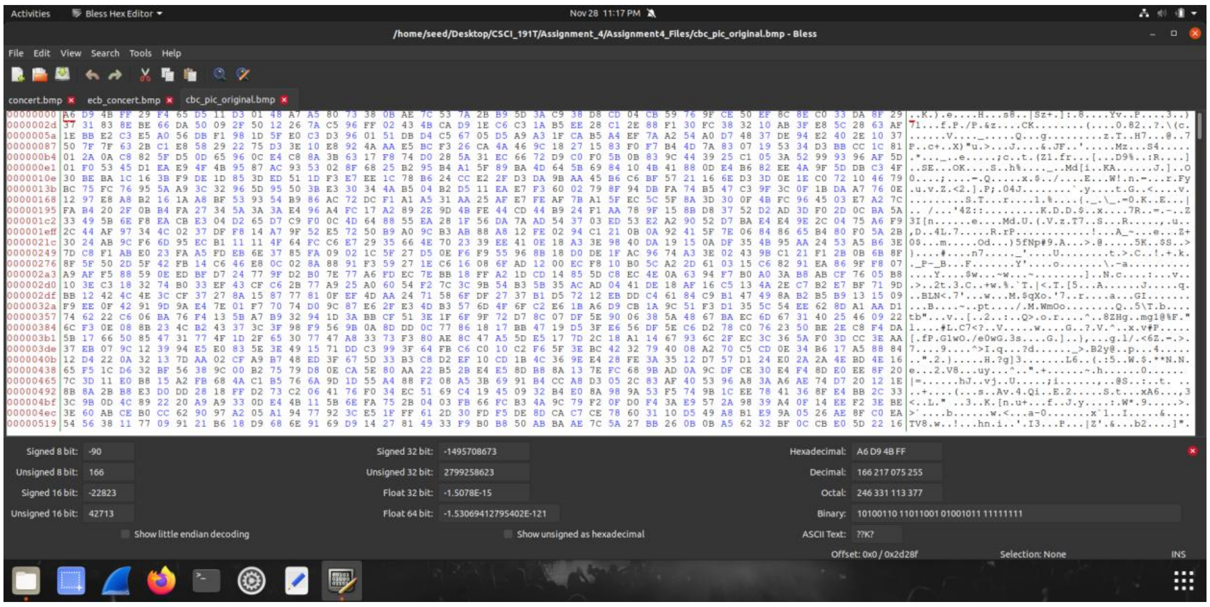
```
Signed 8 bit: 12            Signed 32 bit: 202901202                 Hexadecimal: 0C 18 06 D2
Unsigned 8 bit: 12          Unsigned 32 bit: 202901202               Decimal: 012 024 006 210
Signed 16 bit: 3096         Float 32 bit: 1.171171E-31               Octal: 014 030 006 322
Unsigned 16 bit: 3096       Float 64 bit: 2.0973790921655E-250       Binary: 00001100 00011000 00000110 11010010
      Show little endian decoding      Show unsigned as hexadecimal  ASCII Text: ???
                                                                     Offset: 0x0 / 0xaa8caf    Selection: None    INS
```

Step 2: Using the ECB method, the image in this case is very hard to see but there are some outlines that can be observed on the image. The hex editor also changes once again in the ECB encryption mode for this image.



```
[11/28/22]seed@VM:~/.../Assignment_4$ openssl enc -aes-128-cbc -e -in concert.bmp -out cbc_c
oncert.bmp \-K 00112233445566778899aabbccddee \-iv 0001020334455667689
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[11/28/22]seed@VM:~/.../Assignment_4$ head -c 54 concert.bmp > header
[11/28/22]seed@VM:~/.../Assignment_4$ tail -c +55 cbc_concert.bmp > body
[11/28/22]seed@VM:~/.../Assignment_4$ cat header body > cbc_new_concert.bmp
[11/28/22]seed@VM:~/.../Assignment_4$
```

Activities
Terminal
Nov 28 11:13 PM
seed@VM: ~/.../Assignment_4
cbc_new_concert.bmp
27%
"cbc_new_concert.bmp" selected (11.2 MB)



```
/home/seed/Desktop/CSCI_191T/Assignment_4/Assignment4_Files/cbc_pic_original.bmp - Bless

000003b1 5B 17 66 50 85 47 31 77 4F 1D 2F 65 30 77 47 A8 33 73 F3 80 AE 8C 47 A5 5D E5 17 7D 2C 18 A1 14 67 93 6C 2F EC 3C 36 5A F0 3D CC 3E AA [.fP.G1wO./e0wG.3s....G.]..},...g.l/.<6Z.=.>.
000003de 37 EB 07 9C 12 39 94 E5 E0 83 5E 3E 49 15 71 DD C3 99 3F 64 FB C6 C0 10 C2 F6 5F 3E BC 42 32 79 40 08 A2 70 C5 CD 0E 34 B6 17 A5 88 84 7....9....^>I.q...?d......_>.B2y@..p...4.....
0000040b 12 D4 22 0A 32 13 7D AA 02 CF A9 B7 48 ED 3F 67 5D 33 B3 C8 D2 EF 10 CD 1B 4C 36 9E E4 28 FE 3A 35 12 D7 57 D1 24 E0 2A 2A 4E BD 4E 16 .."..2.}.....H.?g]3.......L6..(.:5..W.$.**N.N.
00000438 65 F5 1C D6 32 BF 56 38 9C 00 B2 75 79 D8 0E CA 5E 80 AA 22 B5 2B E4 E5 8D B8 8A 13 7E FC 68 9B AD 0A 9C DF CE 30 E4 F4 8D E0 EE 8F 20 e...2.V8...uy...^..".+.....~.h.......0......
00000465 7C 3D 11 E0 B8 15 A2 FB 68 4A C1 B5 76 6A 9D 1D 55 A4 88 F2 08 A5 3B 69 91 B4 CC A8 D3 05 2C 83 AF 40 53 96 A8 3A A6 AE 74 D7 20 12 1E |=......hJ..vj..U.....;i.......,..@S..:..t. ..
00000492 8B 8A 2B B8 E3 D0 DD 28 18 FF D2 73 C2 06 41 76 F0 34 EC 51 69 C4 19 45 09 32 B4 E0 8A 98 9A 53 F5 74 9B 1C EE 78 41 36 8F E4 BB 2C 33 ..+....(...s..Av.4.Qi..E.2.....S.t...xA6...,3
000004bf 3C 9B 0D 4C 89 22 20 A9 A9 33 0D E4 4B 11 5B 6E FA 75 2B 04 03 FB 66 FC B3 4A 9C 79 F2 0F D0 F4 3A E9 57 2A 98 39 A4 0F 14 EE F2 3E BE <..L." ..3..K.[n.u+...f..J.y....:.W*.9.....>.
000004ec 3E 60 AB CE B0 CC 62 90 97 A2 05 A1 94 77 92 3C E5 1F FF 61 2D 30 FD F5 DE 8D CA C7 CE 78 60 31 10 D5 49 A8 B1 E9 9A 05 26 AE 8F C0 EA >`....b......w.<...a-0.......x`1..I.....&....
00000519 54 56 38 11 77 09 91 21 B6 18 D9 68 6E 91 69 D9 14 27 81 49 33 F9 B0 B8 50 AB BA AE 7C 5A 27 BB 26 0B 0B A5 62 32 BF 0C CB E0 5D 22 16 TV8.w..!...hn.i..'.I3...P...|Z'.&...b2....]".

Signed 8 bit: -90                Signed 32 bit: -1495708673              Hexadecimal: A6 D9 4B FF
Unsigned 8 bit: 166              Unsigned 32 bit: 2799258623             Decimal: 166 217 075 255
Signed 16 bit: -22823            Float 32 bit: -1.5078E-15               Octal: 246 331 113 377
Unsigned 16 bit: 42713           Float 64 bit: -1.53069412795402E-121    Binary: 10100110 11011001 01001011 11111111
Show little endian decoding      Show unsigned as hexadecimal            ASCII Text: ??K?
                                                                         Offset: 0x0 / 0x2d28f     Selection: None     INS
```

Step 3: Like the previous CBC encryption mode, this mode blocks out the image entirely, therefore not making it viewable even through the image viewer. The hex editor has also changed with the mode of encryption being used.

To conclude, from both images used in Part 1 and Part 2, it can be observed that the ECB mode somewhat partially encrypts the files whereas the CBC mode completely prevents anyone from viewing the image. Despite the EBC mode blurring the image, the image does not show its original colors and makes the image viewer to display it a little differently.