REPORT

1. Can you derive any useful information about the original picture from the encrypted picture? For each mode, describe your observation then explain.

In this experiment, we encrypted the IU logo image using two different encryption modes: ECB and CBC. The objective was to understand how these modes affect the visual characteristics of the encrypted images and whether any information about the original picture can be derived from them.

Original Image Size: **3,858,054 bytes**

ECB Encryption:

- Encrypted Image Size (with --nosalt): **3,858,064 bytes** (**10 bytes more** than the original size)
- Process: We copied the first 54 bytes (bitmap header) from the original image to create a legitimate temp_ecb.bmp file. The remaining bytes were appended from the ECB-encrypted file.

CBC Encryption:

- Encrypted Image Size (with --nosalt): **3,858,064 bytes**
- Process: Like ECB, we copied the first 54 bytes from the original image and appended the remaining bytes from the CBC-encrypted file.

Observations and Explanations:

1. ECB-Encrypted Image (temp_ecb.bmp):

- Visible Outline: In the ECB-encrypted image, we observed that the IU logo outline was still visible. Additionally, some bits of red color were also discernible.
- Explanation: ECB operates on fixed-size blocks of data independently, and identical plaintext blocks result in identical ciphertext blocks. This pattern can reveal details of the original image, leading to the observed outline and color bits. ECB lacks diffusion and is not suitable for image encryption.

2. CBC-Encrypted Image (temp_cbc.bmp):

- No Visibility of Original Image: In the CBC-encrypted image, we observed that the original IU logo image was not visible. Instead, the image appeared as silver, red, and blue dots.
- Explanation: CBC mode XORs each plaintext block with the previous ciphertext block before encryption. This chaining effect ensures that identical plaintext blocks do not produce identical ciphertext blocks, enhancing security. The lack of visibility in the CBC-encrypted image demonstrates its effectiveness in concealing image details.

Conclusion:

The experiment clearly illustrates the differences between ECB and CBC encryption modes. While ECB mode fails to conceal the original image details due to its lack of diffusion, CBC mode effectively hides the image, making it indiscernible. This highlights the importance of choosing a suitable encryption mode for image security, with CBC being a more secure choice in this context.

Recommendation:

For image encryption and confidentiality, it is recommended to use encryption modes like CBC, which provide better security by preventing the visual recognition of the original image.

2.3 a. How much information can you recover by decrypting the corrupted file, if the encryption mode is ECB, CBC, or OFB, respectively? Think about this question before you conduct this task. After your experiments, describe your observation and explain.

Before conducting the task, initial assumptions based on theory

ECB mode encrypts each block of data independently. Therefore, if a single bit in a block is corrupted, it should only affect that specific block, leaving the rest of the data intact. However, the encryption mode lacks diffusion, so identifying patterns in the plaintext could still be possible.

CBC mode uses a chaining mechanism, where each block is XORed with the previous ciphertext block before encryption. If a single bit in a block is corrupted, it can impact the entire chain of blocks, potentially making the entire decryption result less predictable.

OFB mode creates a stream of pseudo-random bits that is XORed with the plaintext. A single bit corruption in the encrypted data can disrupt the entire stream, affecting the entire decryption. The impact could be like CBC mode, potentially making the entire decryption result less predictable.

After the task was conducted, these were the observations

After corrupting a single bit in the encrypted file in ECB mode, it was possible to **recover a recognizable portion of the original text**. This is because ECB mode operates on fixed-size blocks independently, and a single bit corruption in one block only affects that specific block.

In CBC mode, the **corrupted file was less recognizable than in ECB mode**. Corruption in one block affected the entire chain of blocks, making the entire decryption result less predictable.

The OFB mode **resulted in the highest recovery of information after corruption**. Despite a single-bit corruption, most of the original text could be recovered. OFB mode generates a pseudo-random bitstream that XORs with the plaintext, making it less sensitive to local data corruption.

2.3 b. Would any of ECB, CBC, OFB modes serve the purpose of Message Authentication Codes?

None of the ECB, CBC, or OFB modes is suitable for serving as Message Authentication Codes (MACs).

ECB mode is deterministic and lacks the necessary security properties to serve as a Message Authentication Code. It does not provide authenticity or integrity checks, making it vulnerable to various attacks.

While CBC mode provides some level of security, it is not designed to serve as a MAC. To achieve message authentication, dedicated MAC algorithms like HMAC (Hash-based Message Authentication Code) or authenticated encryption modes like GCM (Galois/Counter Mode) should be used.

Like ECB, OFB mode is not designed to serve as a MAC. It lacks the built-in integrity and authenticity checks required for secure message authentication.

In conclusion, none of these encryption techniques are appropriate for Message Authentication Codes; instead, authenticated encryption techniques or specific MAC algorithms should be used to guarantee the authenticity and integrity of messages.