جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering & Technology
Electrical & Computer Engineering Department

ENCS4320-Applied Cryptography

Homework#2

Image Encryption and Decryption Using TEA in ECB and CBC Modes

Instructor:
Ahmad Shawahna

Prepared By:
Abd Khuffash-1200970

Section:
1

Date:
1-6-2024

This project demonstrates the use of the Tiny Encryption Algorithm (TEA) to encrypt and decrypt an image in both Electronic Codebook (ECB) and Cipher Block Chaining (CBC) modes. The process involves converting an image to blocks of data, encrypting and decrypting those blocks, and then reconstructing the image from the encrypted and decrypted blocks.

Libraries Used

- **Pillow (PIL)**: For image processing tasks such as opening, manipulating, and saving image files.
- **NumPy**: For numerical operations and handling arrays.
- **Matplotlib**: For displaying images.
- **Requests**: For downloading images from URLs.
- **io.BytesIO**: For handling image data in memory.

Constants

- **DELTA**: A key schedule constant.
- **SUM_INIT**: The initial sum value for the TEA decryption process.
- **ROUNDS**: Specifies the number of rounds in the TEA encryption and decryption processes.

Functions

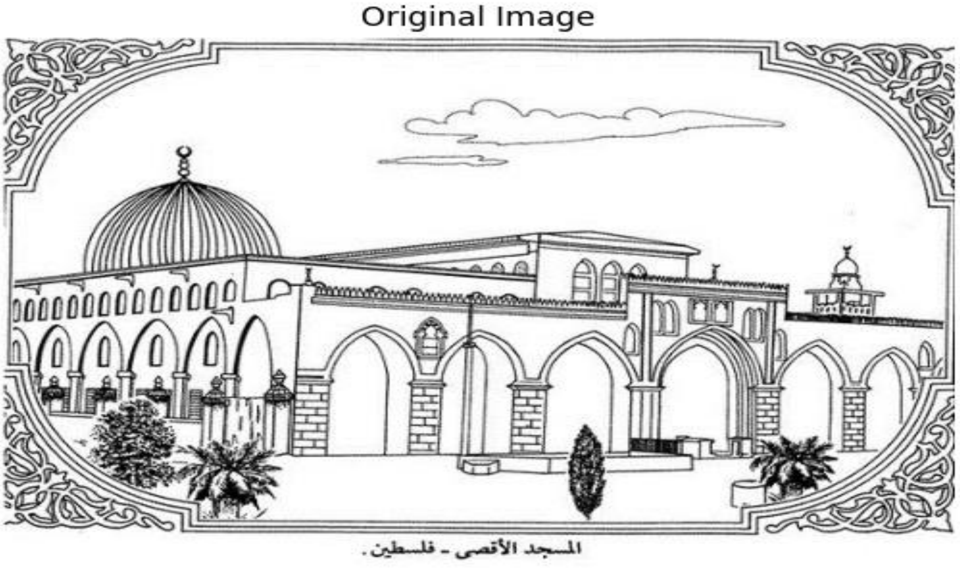
- **tea_encrypt** and **tea_decrypt**: Implement the TEA encryption and decryption algorithms.
- **xor_blocks**: Performs XOR operation on two blocks.
- **encrypt_ecb** and **decrypt_ecb**: Perform ECB mode encryption and decryption.
- **encrypt_cbc** and **decrypt_cbc**: Perform CBC mode encryption and decryption using an initialization vector (IV).
- **image_to_blocks**: Converts an image to a list of (L, R) blocks.
- **blocks_to_image**: Converts a list of (L, R) blocks back into an image.
- **download_image**: Downloads an image from a URL.

Process

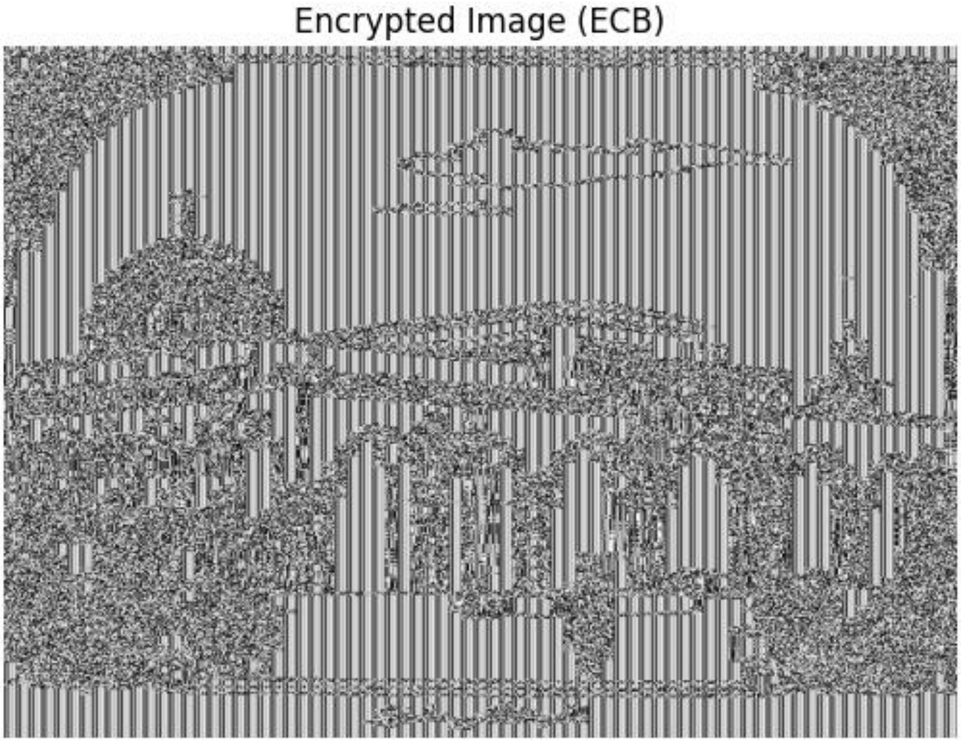
1. **Loading and Displaying the Image**:
   - The user inputs the path or URL to the image.
   - The image is downloaded or loaded from the file system.
   - The image is converted to grayscale and displayed.
2. **Image to Blocks Conversion**:
   - The image is converted into blocks of data suitable for encryption.
3. **User Input for Key and IV**:
   - The user inputs a key and an IV for encryption.
4. **Encryption and Decryption**:
   - The image is encrypted and decrypted using ECB mode.
   - The image is encrypted and decrypted using CBC mode.
5. **Displaying Results**:
   - The original, encrypted, and decrypted images are displayed.

Results

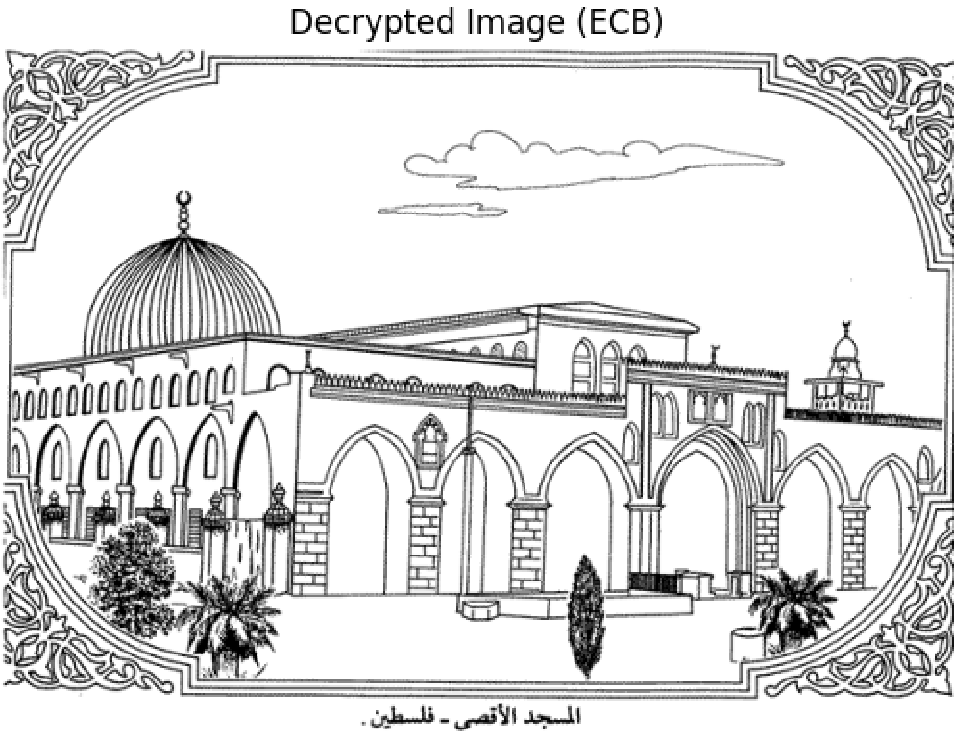
Original Image



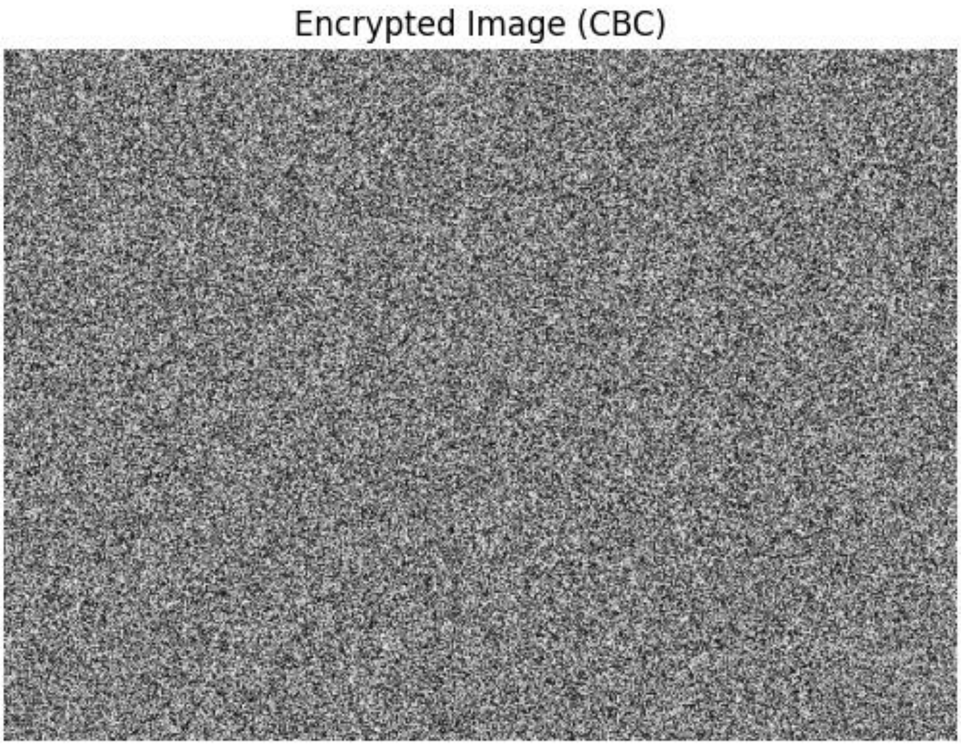
Encrypted Image (ECB)



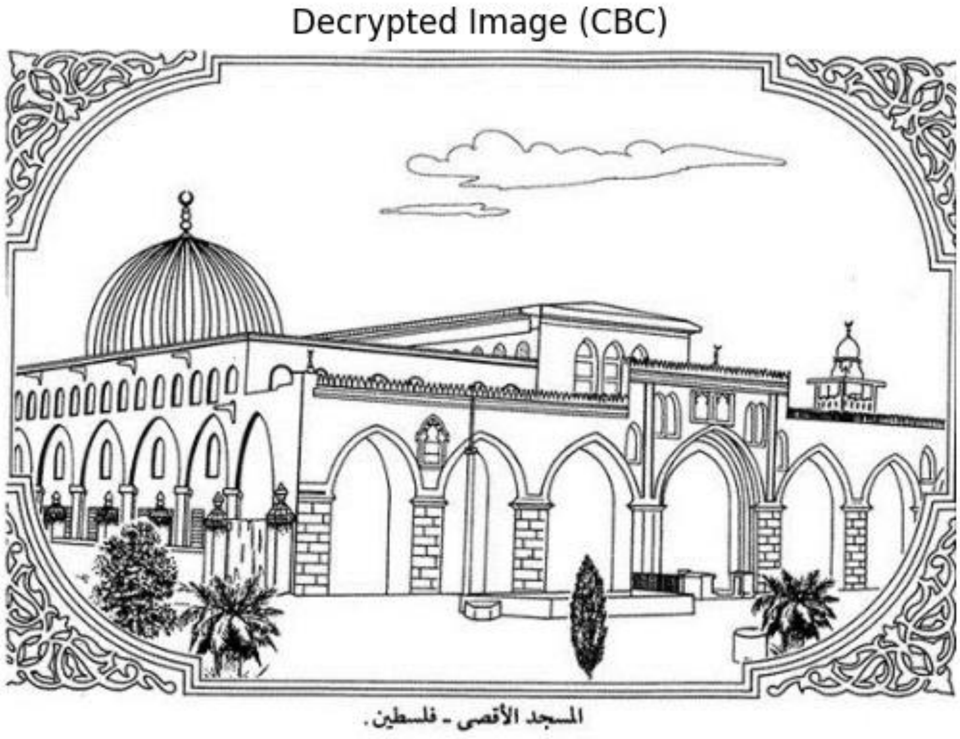
Decrypted Image (ECB)



Encrypted Image (CBC)



Decrypted Image (CBC)



Analysis

- **ECB Mode**:
    - Encryption in ECB mode results in an image that retains some recognizable patterns from the original image, indicating that ECB mode is less secure for image encryption as it does not hide data patterns effectively.
    - Decryption in ECB mode successfully reconstructs the original image, showing that the encryption and decryption processes are working correctly.
- **CBC Mode**:
    - Encryption in CBC mode produces an image where the original patterns are not easily recognizable, demonstrating better security by hiding data patterns more effectively.
    - Decryption in CBC mode also successfully reconstructs the original image, verifying the correctness of the encryption and decryption processes.
    -

In conclusion, this project demonstrates the use of TEA for encrypting and decrypting images in ECB and CBC modes. While ECB mode is simpler, it is less secure due to pattern retention. CBC mode offers better security by effectively masking patterns in the encrypted image. The project highlights the importance of choosing the right encryption mode for securing image data.