CSE 130 – INTRO TO CRYPTOGRAPHY
Spring 2025

Lab 02: Implementation of Block-Cipher Modes of Operation

1 Overview

In this assignment, you are asked to implement and analyze various block-cipher modes of operation in Section 3.6.2 of the Textbook (2nd Ed.), including:

- Electronic Codebook (ECB):
- Cipher Block Chaining (CBC)
- Output Feedback (OFB)
- Counter (CTR)

You will use the **AES-128** block-cipher and encrypt the provided grayscale image to observe the effects of different encryption modes.

2 Specifications

- Block Cipher: AES-128
- Key Length: 128 bits (16 bytes)
- Block Size: 128 bits (16 bytes)
- Modes of Operation: ECB, CBC, OFB, CTR
- Data for Encryption: A grayscale BMP image
- Programming Language: Python (Recommended) or C

3 Tasks

Task 1: Implement and test four block cipher modes of operation using AES-128:

- ECB: Break the plaintext into 16-byte blocks and encrypt each block independently.
- CBC: Use a random 16-byte IV. XOR each plaintext block with the previous ciphertext block before encrypting.
- OFB: Treat the cipher as a keystream generator. Encrypt the IV, then repeatedly encrypt the output to form a keystream.
- CTR: Construct a counter block (nonce + incrementing counter) and encrypt it to generate a keystream. XOR with the plaintext.

Each implementation must correctly: a) Perform encryption and decryption; b) Apply proper padding for block-based modes (e.g., ECB, CBC), and c) Process binary image data while preserving format (using BMP headers and Pillow).

Task 2: Visualize your results on encrypted images. More specifically:

- Encrypt the provided BMP image using each mode.
- Save the encrypted images (e.g., `ecb_encrypted.bmp`) for visual inspection.
- Decrypt and save each encrypted image (e.g., `ecb_decrypted.bmp`) to verify correctness.
- Include the original, encrypted, and decrypted images in your report.

TASK 3: Perform a corruption test to simulate real-world transmission errors as follows:

- Flip a few random bits in the `encrypted` image files.
- Attempt decryption and inspect the visual output.
- Observe and document how each mode responds to corruption:
  - Which modes propagate the error to neighboring blocks?
  - Which modes localize the error?

TASK 4: Conduct a performance benchmark for each mode of operation as follows:

- Calculate and record the encryption and decryption times for the full image (you may want to repeat the process multiple times and calculate the average to get an accurate result).
- Optionally, conduct a scaling benchmark, where you record these timings for the same image at different resolutions.
- Generate a performance comparison table.
- Briefly discuss any observed differences.

TASK 5: Summarize your observations and findings in a report that includes the following:

- Describe the strengths and weaknesses of each mode.
- Discuss practical considerations like IV reuse, parallelizability, and stream vs block behavior.
- Reflect on the results of the corruption and timing tests.

4 Deliverables

- Python (or C) source code: ECB, CBC, OFB, CTR implementations.
- Encrypted and decrypted images for each mode.
- Corrupted image experiments.
- Performance benchmark data.
- A report that includes your analysis and annotated results.