ECE 4301 Final Project Group I

AES-256 Encryption and Decryption on Raspberry Pi with ECB and CBC Modes

AES-256 Encryption and Decryption on Raspberry Pi with ECB and CBC Modes

Objective:

- Implement AES-256 encryption and decryption in Python.
- Support dynamic switching between **ECB** and **CBC** modes.
- Benchmark performance and analyze security implications.

Technical Details:

- **AES-256**:
  - Symmetric block cipher with 256-bit keys.
  - Processes 128-bit blocks in 14 rounds.
- **Modes**:
  - **ECB**: Encrypts blocks independently; prone to pattern leakage.
  - **CBC**: Encrypts blocks with chaining and IV for added security.

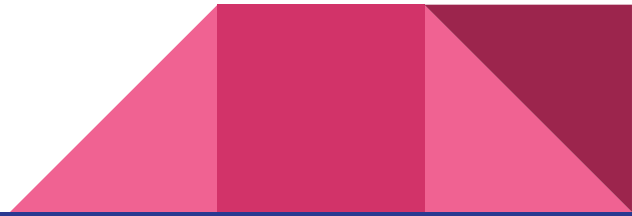
Encryption Modes in Detail

ECB (Electronic Codebook):

- **Mechanism**: Encrypts each 128-bit block independently.
- **Security Flaws**: Repetitive patterns in plaintext result in visible patterns in ciphertext.
- **Use Case**: Demonstration of performance and security limitations.

CBC (Cipher Block Chaining):

- **Mechanism**: XORs each plaintext block with the previous ciphertext block before encryption.
- **Key Feature**: Requires a secure Initialization Vector (IV) for the first block.
- **Advantages**: Eliminates patterns in ciphertext, ensuring better security.
- **Use Case**: Benchmarking against ECB to highlight performance trade-offs.



Implementation Plan

Encryption on Raspberry Pi:

Dynamic mode selection (ECB/CBC).

Encrypt input files (text, images).

Data Transfer:

Send encrypted files to laptop via Python.

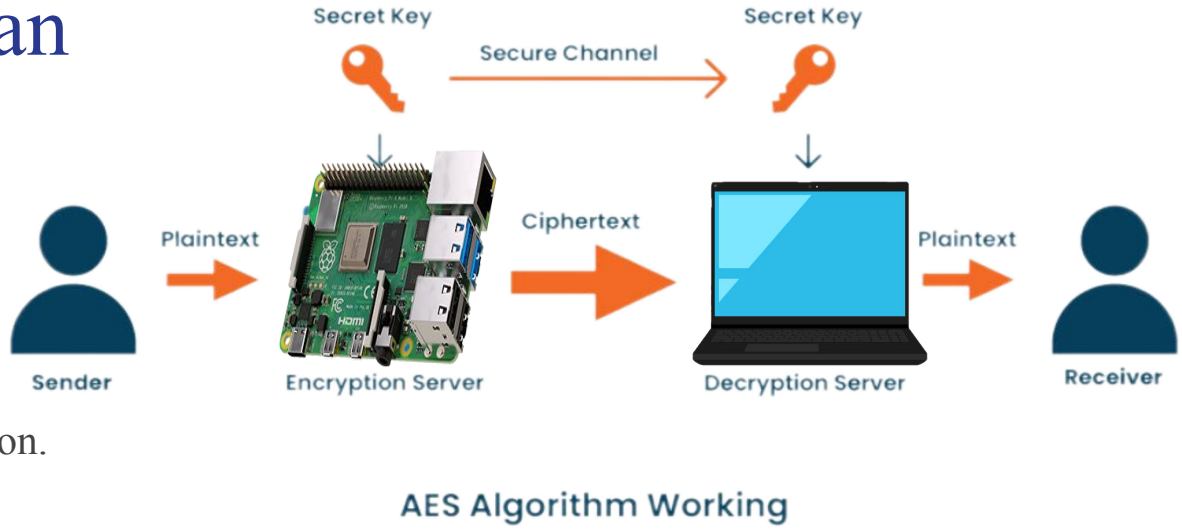
Decryption on Laptop:

Validate output by comparing with original data.

Deliverables:

Measure encryption time, CPU, and memory usage.

Compare performance and security of ECB vs. CBC.



AES Algorithm Working