Implementing and Benchmarking Encryption Modes

Group Members: **Jared Hoyt**

For the final project, I chose to do the following prompt:

"Implement the five modes (ECB, CBC, CFB, OFB, CNT) to encrypt large messages using any block cipher of your choice (DES, AES, etc). Benchmark and compare their performance. Try to use repeated message blocks in different input messages and see the resulting ciphertext. Try also to swap the ciphertext blocks and modify some of the blocks and see the decrypted messages."

Motivation

I was drawn to this prompt not only because of its significant emphasis on programming but also due to the practical application and deep understanding it necessitates. To me, it's more than just an assignment; it's a pathway to expand my skill set in the Python language.

Looking ahead, I envision honing my Python skills to a level where I can comfortably and efficiently implement real-world encryption algorithms. It's not just about becoming adept at Python; it's about understanding the robust encryption algorithms that are fundamental in ensuring data security in today's digital landscape.

I am eager to get to grips with the intricacies of these algorithms, not just in theory but in practice. Understanding how they operate in real scenarios will afford me a comprehensive view of digital security measures, offering knowledge that is both timely and highly relevant in the current era.

Step 1: Preparation

Research

- Understand the five modes of operation (ECB, CBC, CFB, OFB, CTR) and block ciphers (like DES, AES).
- Learn about benchmarking techniques to compare the performance of different encryption modes.

Environment Setup

- Set up a development environment with necessary programming tools (IDE, compilers, etc.).
- Install necessary libraries/packages for cryptography in Python.

Data Preparation

- Prepare a dataset with different large messages to be used during encryption testing.

Step 2: Implementation

Implement Encryption Modes

- ECB (Electronic Codebook)
- CBC (Cipher Block Chaining)
- CFB (Cipher Feedback)
- OFB (Output Feedback)

- CTR (Counter)

Implement Block Ciphers

- DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)

Step 3: Benchmarking

Benchmarking Setup

- Develop a benchmarking strategy to compare the performance of different modes.

Performance Testing

- Run encryption and decryption processes multiple times and record the time taken for each process to get a reliable measure of their performance.

Step 4: Experimentation

Experiment with Repeated Message Blocks

- Encrypt messages with repeated blocks using different modes and observe the ciphertext results.

Ciphertext Modification

Swap Ciphertext Blocks

- Swap different blocks of ciphertext and then decrypt to observe the results.

Modify Ciphertext Blocks

- Make modifications in some blocks and decrypt to see the outcome.

Step 5: Analysis

Data Analysis

- Analyze the benchmark data to identify which mode performs the best in terms of speed and security.
- Analyze the results from the ciphertext experiments to understand how each mode handles different types of input and manipulations.

Documentation

- Document the methodology, the results of your benchmark tests, and the findings from the experimentation phase.

Step 6: Conclusion

Conclusion

- Draw conclusions based on the analysis.

- Offer recommendations for the best modes and ciphers to use for encrypting large messages.

Report

- Compile a comprehensive report presenting the methodology, findings, and conclusions.
- Include visual aids such as charts and graphs to help illustrate the results.

Final Submission

- Make necessary revisions based on the feedback received.
- Submit the final report and ensure to include all the necessary code files and data used in the project.