AES-128 Cryptography Simulator

Contributors:

Muhammad Rehan Siddiqui (ID: 24K-0707)

Abdullah Razzaq (ID: 24K-0691)

Course: Computer Organization and Assembly Language

Instructor: Sir Ghulam Ahmed Burgari

Department of Computer Science, FAST NUCES Karachi

1. Introduction

Data protection has become an essential aspect of modern computing systems. The **Advanced Encryption Standard (AES)** is one of the most widely used symmetric key encryption algorithms for securing digital information.

This project aims to design a **mini AES-128 simulator** entirely in Assembly Language to demonstrate how encryption occurs at the hardware-instruction level. The focus is to show how registers, memory, and arithmetic-logic operations cooperate to transform plain data into cipher text, providing a practical view of how a cryptographic system functions inside a computer.

2. Project Overview

The simulator will implement a simplified version of the AES-128 encryption process. Users will input a small data block and a key. The simulator will then perform essential AES operations such as **SubBytes**, **ShiftRows**, **MixColumns**, and **AddRoundKey**, displaying the transformation of data after each step.

Each stage will reveal how Assembly instructions manipulate bits, bytes, and registers to perform encryption. This step-by-step flow will bridge the gap between high-level cryptography theory and low-level hardware execution.

3. Scope of the Project

Core Features

- Simplified AES-128 encryption with fewer rounds for clarity.
- Visualization of data flow through registers and memory.
- Demonstration of logical and bitwise transformations.
- User interaction through command-line input/output.

Limitations

- No high-level programming languages will be used.

Tools and Technologies

- **Language:** Assembly (x86)
- **Assembler:** MASM with Irvine32 library

4. Learning Outcomes

By completing this project, students will:

1. Understand how cryptographic algorithms operate at the processor level.
2. Learn the role of the **ALU, registers, and memory addressing** in performing encryption.
3. Strengthen understanding of **bitwise and logical operations** in real-world contexts.
4. Gain practical insight into **instruction cycles, branching, and data movement**.
5. Relate computer organization concepts to modern security mechanisms.

5. Concepts Involved (Assembly Perspective)

- **Register Operations:** Storing and manipulating 8-bit or 32-bit data during encryption.
- **Bitwise Instructions:** Using `XOR`, `AND`, `OR`, `SHL`, `SHR` to perform key mixing and substitution.
- **Memory Access:** Reading and writing data blocks representing plaintext and keys.
- **Control Flow:** Looping through AES rounds and managing instruction jumps.
- **Lookup Tables:** Implementing the S-Box for byte substitution.

6. Algorithm Overview (AES-128)

1. **Key Expansion:** Generate round keys from the main 128-bit key.
2. **Initial Round:** XOR plaintext with the first round key.
3. **Main Rounds (3–5 rounds):**
   - **SubBytes:** Replace each byte using the S-Box table.
   - **ShiftRows:** Perform cyclic row shifts in the data block.
   - **MixColumns:** Mix column data using XOR and bitwise operations.
   - **AddRoundKey:** Combine state with round key through XOR.
4. **Final Round:** Repeat SubBytes, ShiftRows, and AddRoundKey.
5. **Output:** Display the final encrypted block.

7. Expected Outcome

The simulator will provide a step-by-step visualization of how a CPU executes cryptographic operations. It will deepen understanding of **Assembly-level computation**, **data manipulation**, and **hardware-driven encryption**.

This project combines theory with practical implementation, making it both **technically insightful and educationally valuable** for Computer Organization and Assembly Language learning.

Submitted by:
Muhammad Rehan Siddiqui (24K-0707)

Abdullah Razzaq (24K-0691)
Department of Computer Science
FAST NUCES Karachi