National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

DEPARTMENT OF COMPUTING

CS-353: Information Security

Class: BSCS-14ABC

Lab 14: Open Ended Lab

CLO-3: Assembles information security solutions by applying key concepts using a variety of tools and techniques

Date: 15th Dec 2025 / 17th Dec 2025

Time: 14:00 - 16:50 / 09:00 - 11:50

Lab Instructor: Mr. Moeed Ahmed

Class Instructor: Dr. Muhammad Ashraf

Lab 14: Open Ended Lab

Introduction:

Advanced Encryption Standard (AES) is a widely used symmetric block cipher that operates on fixed-size blocks of 128 bits and supports key sizes of 128, 192, or 256 bits. AES-128 uses a 128-bit master key and performs 10 rounds of transformation to convert plaintext into ciphertext.

This open-ended lab focuses on a complete, low-level implementation of AES-128, including key scheduling, round transformations, and final encryption output. Students will derive plaintext and keys from their own names, ensuring uniqueness and discouraging plagiarism.

Objectives

In this lab, students will learn:

- Implement AES-128 without using cryptographic libraries
- Convert ASCII text to hexadecimal format correctly
- Apply the AES Key Expansion (Key Schedule) algorithm
- Implement AES round transformations such as SubBytes, ShiftRows, MixColumns and AddRoundKey
- Demonstrate full AES-128 encryption through 10 rounds

Lab Tasks

The lab need to be performed in groups of two members each.
Plaintext (Hex): Use the first student's full name (First Name + Middle Name + Last Name), taking the first 16 characters. Convert each character to its ASCII code, then to hexadecimal.
Master Key (Hex): Use the second student's full name (First Name + Middle Name + Last Name), taking the first 16 characters. Convert each character to its ASCII code, then to hexadecimal.

Task-1: Key Expansion (Key Schedule)
Task-2: Initial AddRoundKey
Task-3: Round 1 Transformations
Task-4: Complete AES-128 Encryption

Constraints:

- *No cryptographic libraries (e.g., OpenSSL, PyCrypto, CryptoJS)*
- *AES tables (S-Box, Rcon) may be hardcoded*
- *All transformations must be student-implemented*

Deliverable:

1. Submit a report which contains individual roles of both group members and console outputs or screenshots *(don't crop any screenshot)* showing:
   - K_0 and K_1
   - Initial AddRoundKey
   - Round 1 intermediate states
   - Final ciphertext
2. Submit your code files .cpp / .java / .py for both tasks
3. Finalize the document in a well structured manner. Save the file with your name and registration number and upload it on LMS under submission link before the deadline.