جامعة الحسين التقنية
Al Hussein Technical University

HTU Course Title: Cryptography

Session 1 and 5: Practical Session on AES-128 with CBC Mode
Exam time: 120 minutes
Date: 22/04/2024
Student name:
Student ID:

Lab Objectives:

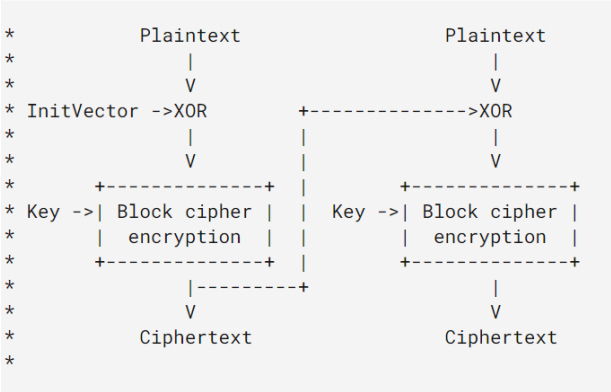
- Implementing AES encryption using either C with OpenSSL or Python with the cryptography library.
- Learn about the Advanced Encryption Standard (AES) and its implementation in Cipher Block Chaining (CBC) mode.
- Gain familiarity with OpenSSL library functions for AES encryption in C or the cryptography library for AES encryption in Python.

Scenario:

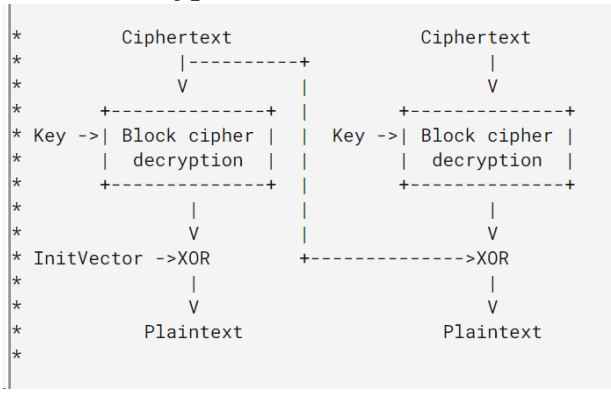
Alice, a computer science student, is learning about encryption techniques. As part of her assignment, she needs to encrypt a confidential message using the Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) mode. She has the option to implement the encryption process in either **C** using the **OpenSSL library** or **Python** using the **cryptography library**.

The following figure ,show the Cipher-block chaining (CBC) cipher mode encryption/decryption, 128 bit key.

- **Encryption:**



- **Decryption:**



جامعة الحسين التقنية
Al Hussein Technical University

Task:

- Alice should input her student name which will be used as the encryption key, and student ID, IV (Initialization Vector).
- Alice inputs the plaintext message she wants to encrypt. The plaintext message provided by Alice will be divided into blocks(such as p1,p2,p3,…pn), with each block containing up to 16 characters. If the length of the plaintext message is not a multiple of 16, padding will be applied to the last block to make it 16 characters long.
    - ✓ For plaintext messages in blocks less than 16 characters, zero padding should be applied to make the plaintext 16 bytes long.
- Alice should save the encrypted output to a file named "encrypted_output.txt".
- Alice should implement the decryption process based on the encrypted output file,and ensuring that the decrypted message matches the original plaintext.
- Alice should change one bit in the encrypted file "encrypted_output.txt" and save the modified file:
    - ✓ Alice implement the decryption process again based on the modified encrypted file.
    - ✓ Note any differences in the decrypted message compared to the original plaintext.

Note:

1-Don't use any ready-made library to build the padding, build it yourself.

2-Do not use the CBC Mode directly from library.

Appendix:

- **Python :**

As part of the lab exercises, you will be required to use specific Python modules from the cryptography library. Understanding these imports is crucial for comprehending how the cryptographic functions operate in your scripts. Below are the necessary imports and a detailed explanation of each to assist you in your lab activities:

```
# Importing cryptographic modules
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.backends import default_backend
```

Explanation of Imports:

1. **Cipher, algorithms, modes**:
    - **Cipher**: This class is used to instantiate encryptor or decryptor objects. It encapsulates algorithms and modes for encryption and decryption processes.
    - **algorithms**: This module includes different cryptographic algorithms. For our lab, we use AES (Advanced Encryption Standard), which is available within this module.
    - **modes**: This module contains the various modes of operation for block ciphers. In this lab, we utilize CBC (Cipher Block Chaining), which is a common mode of operation that provides strong security by chaining together blocks of cipher data.
2. **default_backend**:
    - This function provides a backend entity that implements cryptography algorithms. The backend is responsible for performing the cryptographic operations provided by the library, ensuring they are executed securely and efficiently.

- **C Language :**
  In addition to Python, our lab also utilizes C for implementing cryptographic functions. This section explains the necessary imports and library linking required for C programming, particularly focusing on using OpenSSL for cryptographic operations.

```c
#include <openssl/aes.h>
#include <openssl/rand.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Explanation of Imports and Linking:

1. **OpenSSL Headers**:
   - **<openssl/aes.h>**: This header provides the functions necessary to perform AES encryption and decryption.
   - **<openssl/rand.h>**: This is used for generating cryptographically secure random numbers, essential for creating keys and IVs (Initialization Vectors).
2. **Standard C Libraries**:
   - **<stdio.h>**, **<stdlib.h>**, and **<string.h>**: These are standard libraries in C for handling input/output operations, general utility functions, and string manipulation respectively.

To compile programs that use OpenSSL in C, it is necessary to link the compiler with the OpenSSL libraries. This is typically done by adding -lssl and -lcrypto to your gcc command line.