Birzeit University
Faculty of Engineering and Technology
Department of Electrical and Computer Engineering
ENCS5121 – Information Security and Computer Network Laboratory (Term 1242)
Project (*Phase-2*), Due Sunday, May 11, 2025

A. Document Objectives

This document describes the lab project phase 2, its deliverables, and its grading criteria.

B. Project - Phase II:

Description:

Modify the online guessing game application from **Phase I** to implement encryption and decryption of the data exchanged between the client and server. Use the **AES** symmetric-key encryption system with a **256-bit key** and **CBC** mode. Generate the AES 256-bit key by hashing your student ID using an online SHA256 tool to create a unique 256-bit hashed value. For this phase, you can hardcode the 256-bit key into both the client and server applications. Additionally, ensure that a new Initialization Vector (IV) is generated at the beginning of each game round for use in CBC mode. The client is responsible for generating the IV and sending it to the server. The IV should be 128 bits long to align with the AES plaintext block size and must be generated using a cryptographically secure random number generator (CSRNG) function/method. (You can search for the most suitable Python CSRNG method for your implementation).
If the plaintext data to be sent is less than 128 bits, you may need to pad it accordingly (search online for the correct padding method). Once the IV is generated and the data is encrypted, the client should first send the IV (in plaintext) to the server before transmitting the encrypted data. The server will always expect the first 16 bytes (or 128 bits) of the encrypted data to be the IV used by the client for encryption. The server will then use this IV, along with the hardcoded symmetric key, to decrypt the encrypted data.
Phase II must be your own genuine code. You may use existing libraries/methods/functions but only for AES and CSRNG; the rest of the code needed for **Phase II** must be developed by you.
Make sure to test your modified application preferably using 2 virtual machines. Verify your code by comparing the ciphertext that your application generates against an equivalent ciphertext generated by some of the existing online AES-256-CBC tools (e.g., https://cryptii.com/pipes/aes-encryption).

Note: some of the online AES-256-CBC tools will pad the plaintext even if it was a multiple of 128 bits while others don't. So, if your padding implementation is not similar to that of the online tool, then there will be a mismatch between only the last part of your ciphertext and the ciphertext generated by the online tool.

C. Deliverables

Submit the following items through RITAJ in one .zip file (ID.zip → ***1191615.zip***) before the deadline:

1. Well-commented source code of your implementation, with clear highlights on the sections specifically added in this phase (**.py**).
2. A short video containing the execution part of your application's testing step by step no more than **five minutes**. The video should clearly contain the following:
   a. The unique 256-bit key used by your code.
   b. The plaintext data exchanged between the two sides. Show this information for at least 2 different game rounds.
   c. The corresponding ciphertext along with the IV used for the exchanged data. Show this information for at least 2 different game rounds.
   d. The corresponding verification with an online tool. Make sure to specify the URL of the online tool that you use for verification.
3. A detailed **README** (.txt) file outlining the requirements for running your implementation, along with step-by-step instructions on how to execute and test it.

D. Grading Criteria

The following table summarizes the grading criteria of the lab project phase 2:

Program readability and comments **(.py)**	10 pts.
Step-by-step readme file **(.txt)**	10 pts.
Program executes correctly over a network	20 pts.
Video:	
Screenshot of the unique 256-bit key used by your code	10 pts.
Screenshot of the plaintext data exchanged between the two sides (for at least 2 different game rounds)	10 pts.
Screenshot of the corresponding ciphertext along with the IV used for the exchanged data (for at least 2 different game rounds)	20 pts.
Screenshot of the corresponding verification with an online tool	20 pts.
Total	**100 pts.**

Generally, just by following the guidelines presented in this document, you should get a good score in phase two. However, failing to stick to these guidelines may result in a reduction proportional in magnitude to the deviation.

Good luck, *ENCS5121 Team*