Kingdom of Saudi Arabia
Ministry of Higher Education
Qassim University
College of computer
Compute

المملكة العربية السعودية
وزارة التعليم العالي
جامعة القصيم
كلية الحاسب

Project Report

Coding and Cryptography Course

MATH319 | 471

Supervisor: Dr. Al Anoud Al-Salman

Submission Date:

Students & IDs:

Lubna Aleid	451203449
Dona Alfarraj	451203450
Hams Alshatawa	451203403
Adhwaa Alhudaithy	451203518
Sandra Alomar	451203405

Table of Contents

Secure file storage:

The program provides a secure and user-friendly method for storing files by relying on the AES encryption algorithm, which is one of the strongest and most widely used symmetric encryption methods in data protection worldwide. The program offers users an integrated environment that includes login authentication, file uploading, file encryption, and file retrieval when needed.

The Program Approach:

First, we study and understand the main function and the main purpose of the program, which is to encrypt/decrypt files and protect the user's data. Also, the program must be easy to use. The main goal is to protect the files and user data in a safe and easy way.

- **Authentication:**

  We decided to start with user authentication because it's necessary to identify the identity of the user to protect the data.

- **Registration and Login:**
  - Registration: allows the user to enter the username and password. Then, the program ensures both fields are not empty and the username is not taken before, the program stores the password safely.
  - Login: request from the user to enter the username and password, check if the username is in the USER_FILE, and compare the entered password with the password stored.

- **User Interface:**

  We use Tkinter because it's simple and appropriate for simple programs. We include 1. Login. 2. Register. 3. Dashboard.

- **Encryption and Decryption:**

  We used AES algorithm because it's strong and fast. We use AES-CBC mode.

Project Function:

- **User Interface:**

  User interface (UI) is how users interact with computer systems. It contains username and password, and if the user wants to login or register, it gives the user the choice to encrypt or decrypt a file.
  The UI includes:
  - Login Window: allows the user to enter a username and password, login to their account or register.
  - Register Window: allows new users to create an account.
  - Dashboard Window: allows the logged-in user to encrypt or decrypt files.

- **User Authentication:**
  - The user clicks on registration: when the user creates an account, they should enter the username and password. The program checks that both fields are not empty, and the username does not already exist in the file USER_FILE. Then the password is converted into a hash using SHA-256.
  - The user login: the user should first enter their username and password. The program makes sure that the username already exists in the USER_FILE; if not the user can't enter, and the program sends an error message. Then the program compares the password entered and the password in the file. If both are correct, the user can enter; if they are not, an error message appears, and the program prevents access.
  - Encrypt or decrypt the files only if the user successfully logged in.
  - In the encryption file, the program prompts the user to enter the password for the encryption. In decryption, the programs ask the user to enter the same password as in the encryption process.

- **File Upload:**

  The user selects a file from their device, and the program then prompts them to enter a password for encryption. Then, the program starts to encrypt the file. After the program encrypts the file, the program sends a message that the program finished with the encrypted file.
  If the user doesn't select a file, the program sends a message that says no file selected for encryption.

- **AES Encryption:**

  The program encrypts the file using the AES algorithm in CBC (Cipher Block Chaining) mode. This provides secure encryption by linking each block of data to the previous one.

- **Retrieval and Decryption:**

  If the user chooses to encrypt a file. The program prompts the user to enter the password that is used for encryption. If the password is correct, the program starts to decrypt the file. Then, the user has the original file with a message that says the program decrypted the file successfully.

Integration:

The project's workflow is based on a clear integration between the encryption and decryption functions. The process begins with the user interface, which receives the user's request and calls the appropriate function depending on their choice. When the user opts to encrypt a file, the encryption function generates the necessary data such as the key and IV, then encrypts the file and stores this information at the beginning of the output file. Later, the decryption function relies on the same data produced during encryption: it reads the salt and IV from the encrypted file and uses them to reconstruct the key needed for decryption. In this way, both functions operate in an integrated manner: the output of the encryption serves as the essential input to decryption, ensuring that the process works as a cohesive and secure unit for storing and restoring files.

Privacy:

- **Why Use AES?**

1- very strong encryption key: AES supports keys of 128, 192, or 256 bits.

2- complex algorithm reviewed by a large community.

3- Globally accepted and approved: The algorithm is approved by official security and scientific bodies and is used in government, banking, network, storage systems, and security protocol applications.

4- Efficient and fast execution: Despite its strength, AES is fast (both in programming and hardware) and works effectively even with large files.

- **How can date leakage be prevented?**

The program's operation relies on complete data protection by performing all encryption and decryption operations solely on the user's device, without sending any files or passwords over the internet. This eliminates the possibility of data interception by third parties. The program uses the robust AES-256 algorithm, one of the most widely used and trusted algorithms for data protection globally, for account security, the password itself is not stored; instead, a hash value is stored using SHA-256. This prevents anyone from discovering the password, even if they gain access to the user's profile. When encrypting a file, the system generates a different random value (salt) and initialization vector (IV) for each encryption operation. It then uses the PBKDF2 algorithm to extract a strong key that is difficult to crack using traditional attacks, files are processed as encrypted fragments with padding applied, ensuring that no data is left exposed during the process. Because the program operates entirely locally and does not rely on any external servers, the chances of data leakage are virtually nonexistent.

Challenged Faced:

In this project, we should have a little knowledge of python and user interface. Also, we have some challenges with encrypted or decrypted files, which are:

- **Secure Authentication:**

  We should ensure that usernames and passwords are handled securely. It was a little challenging.

- **Key Derivation and Encryption Logic:**

  We must understand how to generate an encryption key from a user's password. Also, make sure the AES encryption process works correctly and doesn't have any errors.

- **User Interface:**

  Creating a good user interface in a simple way for the user to understand the program was challenging.

- **Error Handling and User Message:**

  We had to make the program user-friendly; we had to make it clear, and we had to send an error message to the user to understand the issue.

- **Testing With Different File Types and Sizes:**

  We tried to make the program know how to encrypt or decrypt with different file formats and different sizes.