CADT
IDT វិទ្យាស្ថានបច្ចេកវិទ្យាឌីជីថល
Institute of Digital Technology

Department of Telecom and Networking

Course: Cryptography

Term 1 | Year 3

Final Project Report

LECTURER: MR. MEAS SOTHEARATH

Name : Loung Sothunvorn

Title : Secure File Locker using AES Encryption

I. Introduction / Background

1.1 Project Overview

Secure File Locker is a desktop-based application designed to protect sensitive files using modern cryptographic techniques. The main goal of this project is to demonstrate how file-level encryption can prevent unauthorized access to confidential data stored on a computer system.

Instead of storing files in plaintext, Secure File Locker encrypts files before they are saved to disk. This ensures that even if the device is lost, stolen, or accessed by an unauthorized user, the file content remains unreadable. The system uses AES-256 encryption with password-based key derivation to provide strong data protection while maintaining ease of use through a graphical user interface.

1.2 Problem Statement

So many users store important files such as personal documents, images, and credentials on their computers without any form of encryption. This creates several security risks:

- Unauthorized users can access sensitive files if a device is lost or stolen
- Malware or malicious software can read files stored in plaintext
- Shared or public computers increase the risk of data exposure
- Users lack simple tools to encrypt files without technical knowledge

Without proper encryption, sensitive data is vulnerable to privacy breaches and data theft.

1.3 Proposed Solution

Secure File Locker addresses these problems by applying strong cryptographic controls at the file level. The proposed solution includes:

- **AES-256-GCM** for encrypting file contents, ensuring confidentiality and integrity
- **Password-Based Key Derivation (PBKDF2 with SHA-256)** to generate secure encryption keys
- **Salt and Nonce generation** to protect against replay and brute-force attacks
- **Graphical User Interface (GUI)** to simplify file selection and encryption operations
- **Separated encrypted and decrypted folders** to prevent accidental file overwriting

II. System Design / Architecture

The Secure File Locker is designed with a modular structure to separate responsibilities clearly. Each part of the system handles a specific task, improving readability, security, and maintainability.

System Components

- **GUI Module (`gui.py`)**
  Handles user interaction such as file selection, password input, and action buttons (Encrypt / Decrypt).
- **Encryption Module (`encrypt.py`)**
  Responsible for encrypting files using AES-256 in GCM mode.
- **Decryption Module (`decrypt.py`)**
  Responsible for decrypting encrypted files back to their original form.
- **Utility Module (`utils.py`)**
  Provides helper functions such as key derivation and output folder management.
- **Main Entry (`main.py`)**
  Starts the application and launches the GUI.

Data Flow

1. User selects a file through the GUI
2. User enters a password
3. System derives a cryptographic key from the password
4. File is encrypted or decrypted
5. Output is saved in a dedicated folder (`encrypted/` or `decrypted/`)

III. Implementation Details

Encryption Algorithm

- **AES-256 (Advanced Encryption Standard)**
- Mode: **GCM (Galois/Counter Mode)**
- Key size: **256 bits**

AES-256 provides strong security and GCM mode ensures both confidentiality and integrity of the data.

Key Derivation

Instead of using the password directly, the system uses **PBKDF2 with SHA-256** to derive a strong cryptographic key. This approach protects against brute-force attacks and enhances security.

File Handling

- Encrypted files are stored in the `encrypted/` folder
- Decrypted files are stored in the `decrypted/` folder
- Output folders are created automatically if they do not exist

Security Features

- Password-based encryption
- No plaintext password storage
- Authentication tag verification during decryption
- Separation of encrypted and decrypted data

IV. Usage Guide

Requirements

- pycryptodome library

Install dependency:

```
pip install pycryptodome
```

How to Run

```
python main.py
```

Steps to Encrypt a File

1. Launch the application
2. Click **Browse** and select a file
3. Enter a password
4. Click **Encrypt**
5. Encrypted file is saved in the `encrypted/` folder

Steps to Decrypt a File

1. Select an encrypted `.enc` file
2. Enter the correct password
3. Click **Decrypt**
4. Decrypted file is saved in the `decrypted/` folder

V. Conclusion and Future Work

This project successfully demonstrates a practical implementation of cryptography through a Secure File Locker application. By using AES-256 encryption and password-based key derivation, the system provides strong protection for sensitive files while remaining easy to use through a graphical interface.

The project shows how encryption can prevent unauthorized access to data, even if files are stolen or copied. The modular design also makes the system easy to maintain and extend.

Future Improvement

1. I will add password with strong password to make it more secure
2. Improve Gui to make it more better and easier for user to use
3. Add log monitoring for encrypt and decrypt file
4. Add attempt for input incorrect password

VI. References

1. NIST – Advanced Encryption Standard (AES)
2. PyCryptodome Documentation
3. OWASP Cryptographic Storage Guidelines