

Secure File Sharing System Report

Target: Secure File Sharing System (Flask + AES Encryption)

Tools Used: Python Flask, PyCryptodome, HTML/CSS/JavaScript

Submitted By: Gourav Swaroop

Date: 26/08/2025

Introduction

The Secure File Sharing System (SFSS) is designed to facilitate safe uploading and downloading of files while ensuring their confidentiality. By implementing AES encryption, the system protects data both during storage and transfer. This approach reflects real-world requirements in industries such as healthcare, legal, and corporate sectors, where sensitive information must be securely handled.

Abstract

This web-based system encrypts files before storage and decrypts them upon retrieval. Uploaded files are converted into an encrypted .enc format, ensuring they remain unreadable without proper access. When users download the files, they are securely decrypted back to their original format. The project demonstrates secure file management, AES encryption, and seamless integration with a Flask backend.

Tools Used

- **Python Flask:** Backend framework enabling secure file uploads and downloads.
 - **PyCryptodome:** Provides AES encryption and decryption functionality.
 - **HTML, CSS, JavaScript:** Frontend interface for user interactions.
 - **VS Code:** Integrated development environment for project implementation.
-

Steps Involved in Building the Project

1. Designed a Flask-based web application with a homepage, upload, and download pages.
2. Implemented AES encryption to transform files into an encrypted .enc format before storage.
3. Stored encrypted files securely in the /uploads/ directory.

4. Developed a decryption feature to restore files to their original format during download.
 5. Verified file integrity by comparing the original and decrypted versions.
 6. Implemented secure key handling to prevent unauthorized access.
 7. **Screenshots of the application, demonstrating file upload, encryption, and download processes, are available in the project repository for reference.**
-

Conclusion

The Secure File Sharing System was successfully implemented, ensuring files remain protected during both storage and transfer. This project demonstrates practical skills in cryptography, backend development, and secure data management. Future enhancements could include user authentication, role-based access control, and integration with cloud storage platforms to further strengthen security and accessibility.