Enterprise Secure File Storage System using AES-256-GCM Encryption

Internship Project Report

Prepared by:

Shusanta Bargayary

Institute:

Central Institute of Technology, Kokrajhar

Date of Submission:

27 October, 2025

1. Introduction

Data security plays a vital role in modern computing systems. With the exponential growth of digital information, ensuring confidentiality, integrity, and availability of data has become crucial. This project demonstrates an **Enterprise Secure File Storage System** that employs **AES-256-GCM encryption** to protect files against unauthorized access and tampering.

The system integrates cryptographic techniques, hashing mechanisms, and database management to ensure complete data protection and recoverability. It provides a user-friendly interface to perform encryption, decryption, and metadata tracking efficiently.

2. Objective

The primary objective of this project is to design and implement a secure file storage application that:

- Encrypts and decrypts files using AES-256-GCM for confidentiality and authenticity.
- Verifies file integrity through SHA-256 hashing.
- Logs all activities in an SQLite database.
- Provides a GUI for easy interaction and monitoring.

3. Tools and Technologies Used

Component	**Description**
Programming Language	Python 3
GUI Framework	Tkinter
Database	SQLite3
Encryption Algorithm	AES-256-GCM (Cryptography Library)
Hash Function	SHA-256
Threading	Python Threading Module

Table 3.1: Tools and Technologies Used

4. Implementation Details

The system architecture follows a modular approach:

1. **Key Management:** A secure 256-bit AES key is generated and stored in a protected file with restricted permissions.
2. **Encryption Process:** The selected file is encrypted in chunks using AES-GCM mode to handle large files. A random nonce ensures uniqueness.
3. **Decryption Process:** The encrypted file is decrypted using the stored key, restoring the original content and verifying authenticity.
4. **Integrity Verification:** Each file is hashed using SHA-256 to detect any modification.
5. **Logging:** The SQLite database records each operation (filename, hashes, nonce, timestamp) for accountability.
6. **User Interface:** Built using Tkinter, it includes progress bars, real-time status updates, and a logs viewer.

4.1. System Workflow Diagram

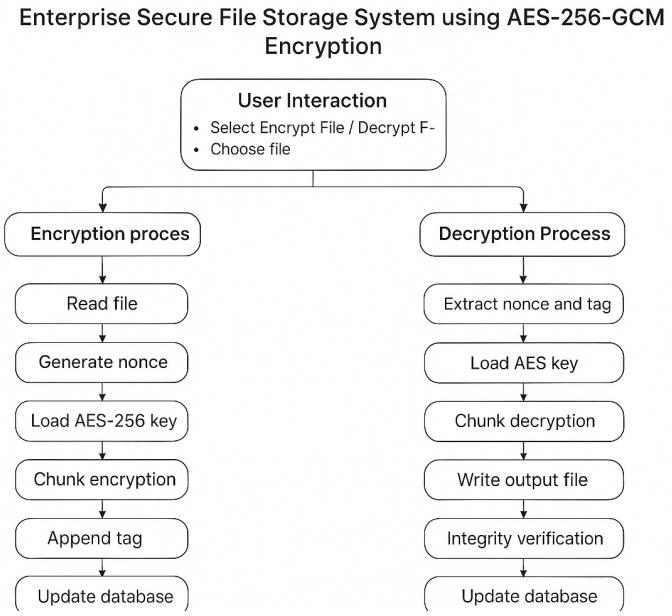


Figure 4.1: Overall System Workflow

5. Key Features

- AES-256-GCM encryption with authentication tag verification.
- Automatic database schema upgrades for compatibility.
- Real-time progress bar and percentage updates.
- Secure key management with file permission control.
- SHA-256 integrity verification of files.
- Responsive Tkinter interface with activity log viewer.

6. Results and Output

The system efficiently encrypts and decrypts files of any size without data loss. The SQLite database successfully maintains a log of each operation. The interface provides smooth feedback throughout the encryption and decryption processes.

6.1. Screenshots and Observations

1. Application Interface

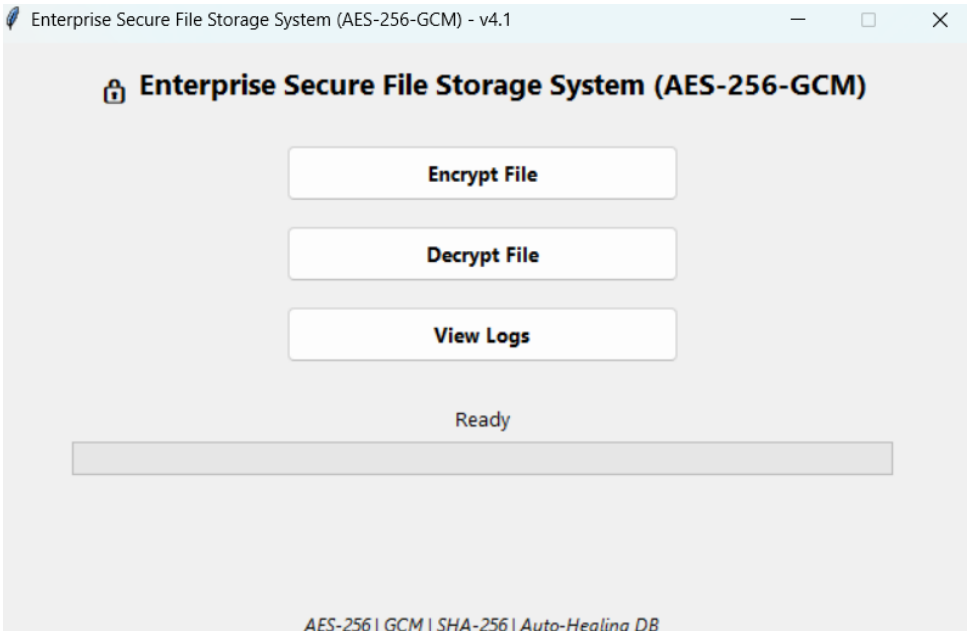


Figure 6.1: Application GUI on Launch

2. Encryption Process

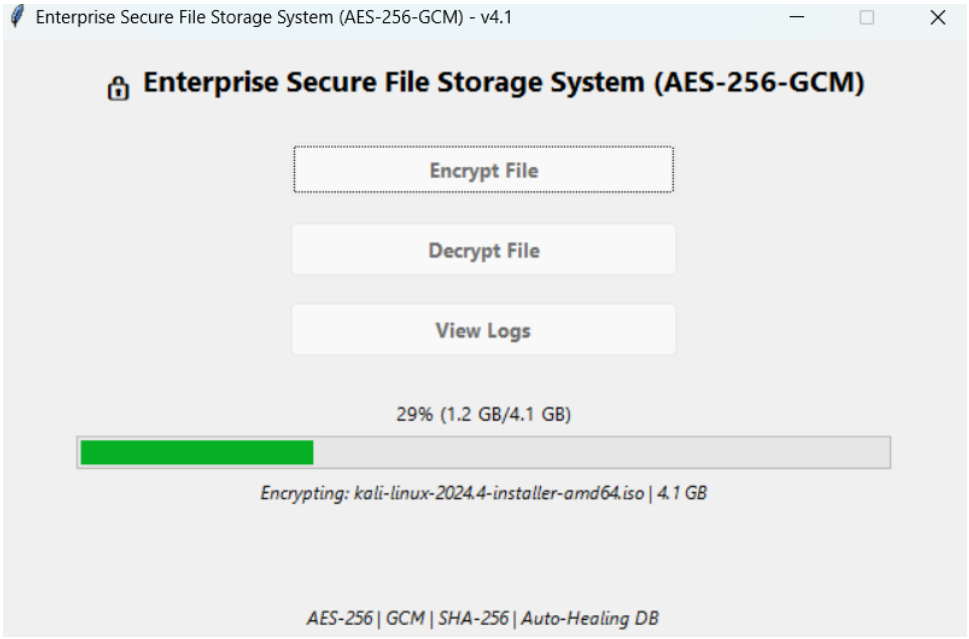


Figure 6.2: Encryption Progress with Real-Time Status

3. Encryption Success Message

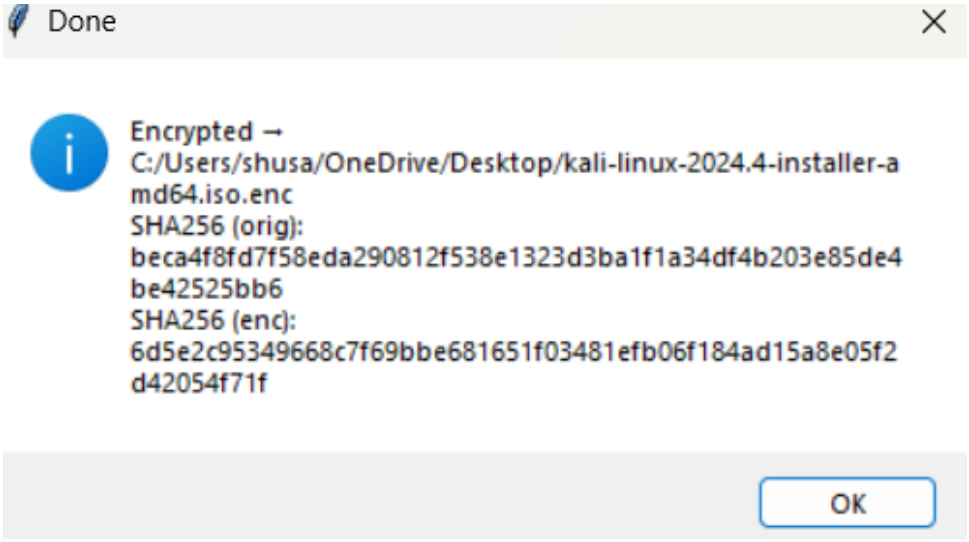


Figure 6.3: Confirmation Dialog after Successful Encryption

4. Decryption Process

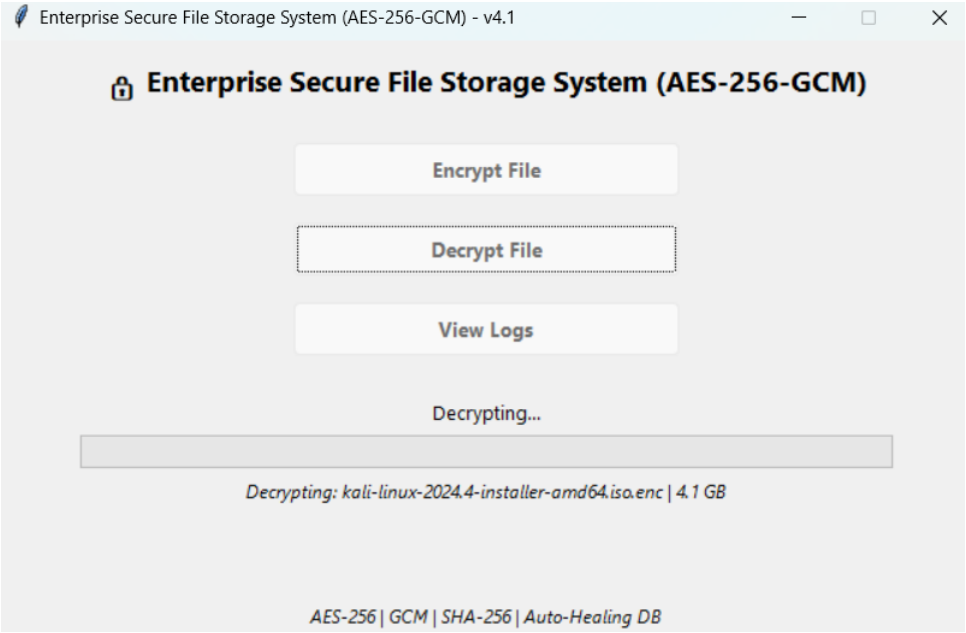
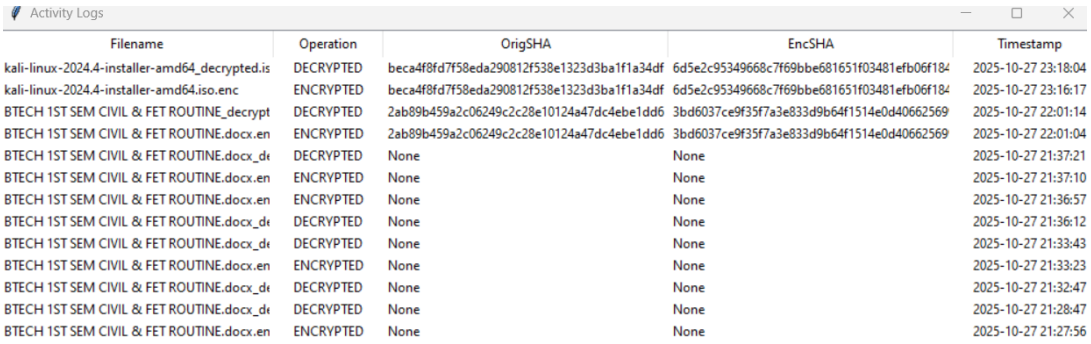


Figure 6.4: Decryption in Progress with Integrity Verification

5. Logs Viewer



Filename	Operation	OrigSHA	EncSHA	Timestamp
kali-linux-2024.4-installer-amd64_decrypted.is	DECRYPTED	beca4f8fd7f58eda290812f538e1323d3ba1f1a34df	6d5e2c95349668c7f69bbe681651f03481efb06f184	2025-10-27 23:18:04
kali-linux-2024.4-installer-amd64.iso.enc	ENCRYPTED	beca4f8fd7f58eda290812f538e1323d3ba1f1a34df	6d5e2c95349668c7f69bbe681651f03481efb06f184	2025-10-27 23:16:17
BTECH 1ST SEM CIVIL & FET ROUTINE_decrypt	DECRYPTED	2ab89b459a2c06249c2c28e10124a47dc4ebe1dd6	3bd6037ce9f35f7a3e833d9b64f1514e0d40662569	2025-10-27 22:01:14
BTECH 1ST SEM CIVIL & FET ROUTINE.docx.en	ENCRYPTED	2ab89b459a2c06249c2c28e10124a47dc4ebe1dd6	3bd6037ce9f35f7a3e833d9b64f1514e0d40662569	2025-10-27 22:01:04
BTECH 1ST SEM CIVIL & FET ROUTINE.docx_d	DECRYPTED	None	None	2025-10-27 21:37:21
BTECH 1ST SEM CIVIL & FET ROUTINE.docx.en	ENCRYPTED	None	None	2025-10-27 21:37:10
BTECH 1ST SEM CIVIL & FET ROUTINE.docx.en	ENCRYPTED	None	None	2025-10-27 21:36:57
BTECH 1ST SEM CIVIL & FET ROUTINE.docx_d	DECRYPTED	None	None	2025-10-27 21:36:12
BTECH 1ST SEM CIVIL & FET ROUTINE.docx_d	DECRYPTED	None	None	2025-10-27 21:33:43
BTECH 1ST SEM CIVIL & FET ROUTINE.docx.en	ENCRYPTED	None	None	2025-10-27 21:33:23
BTECH 1ST SEM CIVIL & FET ROUTINE.docx_d	DECRYPTED	None	None	2025-10-27 21:32:47
BTECH 1ST SEM CIVIL & FET ROUTINE.docx_d	DECRYPTED	None	None	2025-10-27 21:28:47
BTECH 1ST SEM CIVIL & FET ROUTINE.docx.en	ENCRYPTED	None	None	2025-10-27 21:27:56

Figure 6.5: Operation Logs Displayed in the SQLite Log Viewer

7. Conclusion

The project successfully demonstrates a secure, reliable, and efficient file storage system using AES-256-GCM encryption. The implementation of advanced cryptographic techniques and secure key management ensures high-level data protection. The system can be extended in the future with cloud integration and multi-user access control for enterprise-level deployments.