Project Report: Secure File Vault Pro

Title: Secure File Storage System with AES
Technology: Python, PyQt5, Cryptography
Developer: Vishal
Duration: Internship Project – Elevate Labs

1. Introduction

In the digital era, securing confidential files from unauthorized access is a major challenge. Sensitive information such as financial documents, passwords, and personal data requires strong protection mechanisms. This project, SecureFileVaultPro, is designed to provide a local encryption and decryption system that ensures data confidentiality and integrity using advanced AES-256 encryption. It offers an intuitive graphical user interface (GUI) for file operations and secure storage.

2. Abstract

SecureFileVaultPro is a Python-based secure file storage system that allows users to encrypt and decrypt files locally using AES-256 encryption in CTR mode. The system derives encryption keys from a master password using PBKDF2, ensuring robust key management. The application verifies file integrity using HMAC-SHA256 to detect any tampering. The GUI, developed in PyQt5, provides an easy-to-use interface for file selection, encryption, decryption, and log viewing. Users can also view encrypted and decrypted files directly within the app.

3. Tools and Technologies Used

- Programming Language: Python 3.x
- Libraries:
  - PyQt5 – for professional GUI design
  - cryptography – for AES encryption/decryption and HMAC verification
  - json, os, base64 – for file operations and encoding
- IDE: Visual Studio Code / PyCharm
- Storage: Local file system
- Encryption Algorithm: AES-256 (CTR Mode)
- Key Derivation: PBKDF2 with SHA-256 and random salt

4. Steps Involved in Building the Project

1. Design Phase:
   - Defined project structure with app.py, crypto_backend.py, and utils.py.
   - Designed GUI layout using PyQt5 with tabs for encryption, logs, and file viewing.
2. Implementation Phase:
   - Developed AES-based encryption/decryption logic in crypto_backend.py using streaming to handle large files.

- Implemented secure key derivation using PBKDF2.
- Added HMAC verification to detect tampering.

3. GUI Development:
- Built a modern interface in app.py with master password authentication.
- Integrated progress bar, file selector, and activity logs.

4. Logging and Metadata:
- Implemented automatic logging of all activities and file metadata using utils.py.

5. Testing Phase:
- Verified encryption-decryption correctness using sample text and image files.
- Validated integrity check using intentionally modified files.

5. Conclusion

SecureFileVaultPro provides an efficient and reliable solution for protecting local files with AES-256 encryption. It combines security, usability, and transparency through an easy-to-navigate GUI and real-time feedback. This project demonstrates practical implementation of cryptographic principles and can be extended to include cloud synchronization or biometric authentication in the future.