Secure File Share: AES-Encrypted File Transfer System

- By Amithabh D.K

Objective

Build a secure web application for encrypting/decrypting files using AES-256. Files are password-protected, shareable via time-limited links, and automatically deleted after 24 hours.

Tech Stack & Tools Used

1. Python Flask - Backend framework for routing and logic
2. PyCryptodome - AES-256-CBC encryption/decryption library
3. HTML/CSS/JavaScript - Frontend UI with responsive design
4. Font Awesome - Icons for UI elements
5. Werkzeug - Secure file handling and upload validation
6. Git & GitHub - Version control and collaboration
7. Socket Programming - Auto-detection of local IP for network sharing

Security Features Implemented:

Feature	Description
AES-256-CBC Encryption	Military-grade file encryption with unique IV/salt per file
PBKDF2 Key Derivation	100,000 iterations of SHA-256 to strengthen passwords
SHA-256 Integrity Check	Verifies file integrity before/after decryption
24-Hour Auto-Expiry	Background thread deletes files after 24 hours
Brute-Force Protection	Blocks after 3 failed password attempts
File Type Whitelist	Only allows safe extensions (e.g., .txt, .pdf, .png)
Secure Sessions	Encrypted session management for temporary data

Key Functionality

1. File Upload

→ Password-protected encryption (min. 6 characters)

→ Automatic generation of shareable link with file ID

Secure File Share: AES-Encrypted File Transfer System

- By Amithabh D.K

2. Secure Download

→ Password-required decryption

→ Text file preview for supported formats

→ Download countdown timer showing expiry

3. Shareable Links

→ Auto-generated URLs with pre-filled file IDs

→ Works across local networks

How to Run

```
# Step 1: Clone the repository
git clone https://github.com/Amithabh0314/FUTURE_CS_03.git
cd FUTURE_CS_03

# Step 2: Set up a virtual environment
python3 -m venv venv
source venv/bin/activate   # On Windows: venv\Scripts\activate

# Step 3: Install dependencies
pip install -r requirements.txt

# Step 4: Run the Flask app with SSL
python app.py
  Local: http://localhost:5000
  Network: http://<your-local-ip>:5000
```

Secure File Share: AES-Encrypted File Transfer System

- By Amithabh D.K

Key Outcomes & Learnings

→ Implemented end-to-end file encryption with zero-knowledge passwords

→ Solved real-world challenges:

→ Cross-network accessibility using socket-based IP detection

→ Secure memory management for decrypted files

→ Applied cryptographic best practices:

→ Unique IV/salt per file

→ PKCS#7 padding

→ Key stretching via PBKDF2

Limitations & Future Upgrades:

Limitation	Planned Enhancement
No user accounts	Add OAuth/login system
HTTP-only in development	Deploy with HTTPS (Nginx + Let's Encrypt)
No activity logs	Implement audit trails
Manual IP sharing	Add QR code generator for mobile sharing
Limited text preview	Support more formats (Markdown, PDF preview)

Conclusion:

This project demonstrates practical AES-256 implementation for secure file transfers. By combining Flask's simplicity with robust cryptography, it provides a foundation for secure data sharing while adhering to OWASP standards. Future iterations will expand usability and enterprise features.

Developed by: Amithabh D.K

GitHub: https://github.com/Amithabh0314/FUTURE_CS_03

Deployment: Local network (Kali Linux)