University of Engineering and Technology Lahore
Department of Computer Science

Muhammad Abubakar 2022-CS-188
AbdulQayoum Rajpoot 2022-CS-208

AI-Powered Brain Tumor Detection and Knowledge Assistant

In today's digital age, secure file transfer is essential to ensure confidentiality and integrity during data exchange. This project presents a secure file transfer system that uses AES encryption and hashing to protect the transmitted files from unauthorized access and tampering. The objective is to provide a reliable solution for secure file sharing with an easy-to-use interface.

Objective:

To securely transfer files between devices while ensuring data integrity and confidentiality.

Problem Solved:

Addresses the need for secure file sharing by combining encryption and hashing techniques.

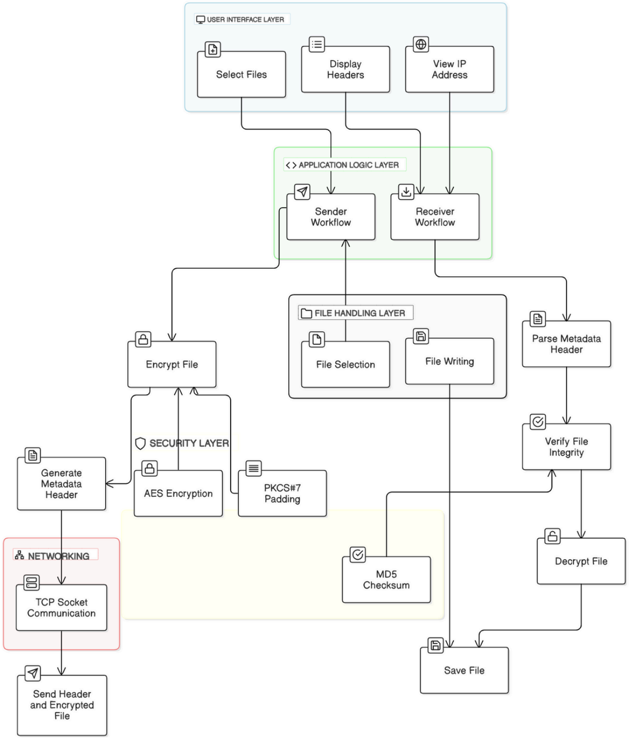
Features

- Two operation modes: Sender and Receiver
- AES encryption (AES-CBC) for secure data transmission.
- MD5 checksum for file integrity verification.
- User-friendly GUI designed with Tkinter.
- Support for all file types.
- Displays detailed file header for transparency.

System Architecture

This system enables secure and reliable file transfer over a network by combining modern cryptographic techniques with robust networking protocols. The architecture is divided into distinct layers, each handling specific functionalities, ensuring modularity and clarity in design.

System Architecture Flowchart

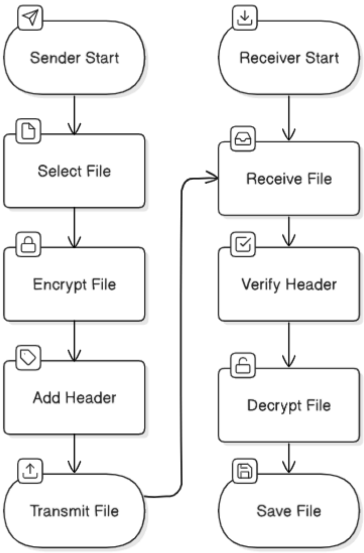


Conclusion

The project successfully fulfills its objective of providing a secure and reliable file transfer mechanism. The integration of AES encryption and MD5 hashing ensures confidentiality and data integrity during the transfer. The system is user-friendly and can be easily extended for additional features.

Workflow

File Transmission Workflow



Libraries/Technologies Used:

- **Python Libraries:**
  - socket for network communication.
  - hashlib for checksum verification.
  - PyCryptodome for AES encryption.
  - tkinter for GUI design.

Protocols and Standards:

- TCP: Used for reliable file transfer.
- AES-CBC: Encryption algorithm for data confidentiality.
- Padding: PKCS#7 ensures data fits the AES block size.

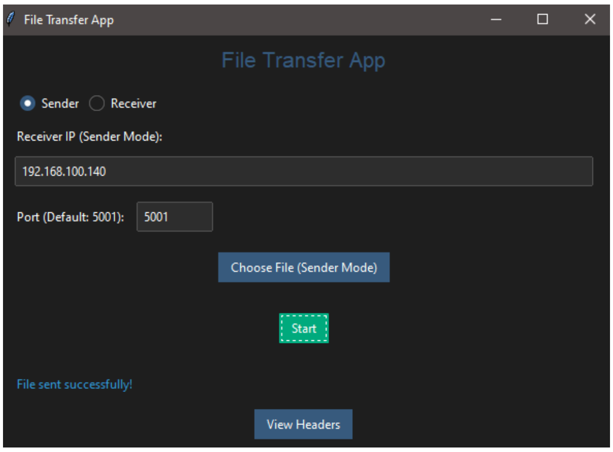
Hashing:

- MD5 checksum validates file integrity during transmission.

Header Metadata:

- Fields: File name, size, source IP, destination IP, timestamp, checksum, etc.

Results



File Transfer App

File Transfer App

● Sender ○ Receiver

Receiver IP (Sender Mode): 192.168.100.140

Port (Default: 5001): 5001

File sent successfully!