SecureShare: A Novel Approach for Secure File Sharing with Integrated Malware Protection

Akash Wagh,Nikhil Wagh,Akanksha Bankar,Sharvari Alsunde,Mrs.Vaidehi Khatvakar

Pune wagholi,412207India

A B S T R A C T

Secure File Share is an Android-based application designed to enable fast, private, and fully secure peer-to-peer (P2P) file transfer without relying on internet connectivity or centralized servers. The system leverages WiFi Direct technology to establish direct communication between devices, ensuring highspeed data transmission while maintaining complete user control over shared data.

To ensure strong security, the application integrates AES-256-GCM encryption for file protection and RSA-2048 for secure session key exchange. Sensitive cryptographic keys are stored using the Android Keystore system, providing hardware-backed security. Additionally, biometric authentication mechanisms such as fingerprint and face recognition are implemented to restrict unauthorized access. Data integrity is verified using SHA-256 hashing, ensuring that files are received without tampering or corruption.

The application also includes features like QR code-based device pairing for quick and secure trust establishment, batch file transfers, real-time progress monitoring, and smart compression to optimize performance. A local SQLite database is used to maintain a detailed transfer history, enabling efficient tracking and management of shared files.

Overall, Secure File Share provides a reliable, secure, and efficient solution for offline file sharing, addressing privacy concerns and eliminating dependency on cloud-based systems while ensuring data confidentiality and integrity

Keywords: Secure File Sharing, Peer-to-Peer (P2P) Communication, WiFi
Direct, AES-256-GCM Encryption, RSA-2048 Key Exchange, Android
Keystore, Biometric Authentication, SHA-256 Integrity Verification, QR Code Pairing, Offline File Transfer, Data Privacy, Secure Deletion, Android Application, SQLite Database, End-to-End Encryption

1. Introduction

In today's digital world, file sharing has become a basic requirement for both personal and professional use. Most existing file transfer solutions depend heavily on internet connectivity or cloud-based platforms, which can raise concerns related to data privacy, security, and third-party access. Users often have limited control over how their data is stored, transmitted, or accessed, making sensitive information vulnerable to breaches or unauthorized use.

To address these challenges, the Secure File Share application is designed as a peer-to-peer (P2P) file transfer system that works without requiring an active internet connection. By using WiFi Direct technology, the application enables direct communication between devices, ensuring faster transfer speeds and eliminating dependency on external servers. This approach not only improves performance but also enhances user privacy by keeping data strictly between the sender and receiver.

Security is a core focus of the system. The application implements advanced cryptographic techniques, including AES-256-GCM for file encryption and RSA-2048 for secure key exchange. Sensitive keys are securely stored using the Android Keystore system, while biometric authentication adds an
additional layer of protection against unauthorized access. Furthermore, SHA-256 hashing is used to verify file integrity, ensuring that transferred data remains unchanged and authentic.

The application also emphasizes usability and efficiency through features such as QR code-based device pairing, batch file transfers, realtime progress tracking, and local storage of transfer history using SQLite. These features make the system not only secure but also userfriendly and practical for everyday use.

Overall, Secure File Share aims to provide a reliable, fast, and secure alternative to traditional file sharing methods by combining strong encryption, offline capabilities, and seamless user experience into a single Android application.

2. Literature Survey

The rapid growth of mobile devices and the increasing demand for fast and secure data transfer have led to extensive research in peer-topeer (P2P) file sharing systems, especially on Android platforms. Traditional file sharing methods rely on centralized servers or internetbased services, which introduce latency, privacy concerns, and dependency on third-party infrastructure. As a result, researchers have explored decentralized communication technologies such as Wi-Fi Direct for efficient device-to-device data exchange.

Wi-Fi Direct has emerged as a key technology for P2P communication, allowing devices to connect directly without requiring a wireless access point. It enables high-speed data transfer and supports applications such as file sharing, gaming, and resource sharing. Studies show that Wi-Fi Direct provides significantly higher throughput and better performance for large file transfers compared to Bluetooth, although it may consume more power . Its ability to establish direct connections makes it highly suitable for offline file sharing applications.

Several research works have focused on the architecture and implementation of Wi-Fi Direct in Android systems. The Android WiFi

P2P framework allows devices to discover peers, establish secure connections, and exchange data efficiently through APIs such as WifiP2pManager . Further studies analyze its internal architecture across different Android layers, including hardware, kernel, and application frameworks, highlighting its flexibility and scalability for mobile applications .

In addition to performance, security has been a major concern in existing file sharing applications. Research indicates that many popular P2P file sharing apps prioritize usability over security, leading to vulnerabilities and potential data leaks . To address these issues, secure communication models have been proposed using hybrid encryption techniques that combine symmetric and asymmetric cryptography for secure key exchange and data protection .

Moreover, studies on peer-to-peer protocols demonstrate that decentralized file sharing systems eliminate the need for centralized servers and improve reliability and scalability. These systems allow devices to dynamically connect and exchange data within a network, making them suitable for offline environments and emergency communication scenarios . However, limitations such as one-hop communication and lack of multi-device routing in standard Wi-Fi Direct implementations have also been identified, leading to ongoing research in multi-hop networking solutions .

Overall, the literature highlights that while existing technologies like Wi-Fi Direct provide a strong foundation for high-speed P2P file sharing, challenges remain in terms of security, scalability, and usability. These gaps motivate the development of secure, efficient, and userfriendly file sharing systems that integrate advanced encryption, reliable communication protocols, and improved user experience, as proposed in this project.

3. System Architecture

The Secure File Share system follows a modular, layered architecture designed to ensure scalability, security, and efficient data transfer. The architecture is divided into multiple functional layers, each responsible for a specific task, enabling smooth interaction between user interface, security mechanisms, and network communication.

1. Presentation Layer (UI Layer)

This layer handles all user interactions and provides an intuitive interface based on Material Design principles. It includes activities such as Dashboard, Device Pairing, File Selection, Transfer Progress, History, and Settings. The UI layer communicates with backend components to initiate pairing, start file transfers, and display real-time updates.

2. Application Logic Layer

This layer acts as the core controller of the system. It manages workflows such as device pairing, file selection, encryption initiation, and transfer handling. It coordinates between different modules like security, networking, and data storage to ensure proper execution of operations.

3. Security Layer

Security is a critical component of the architecture. This layer includes:

- **EncryptionManager**: Handles AES-256-GCM encryption and decryption of files.
- **Key Management**: Uses RSA-2048 for secure session key exchange.
- **Android Keystore**: Stores private keys securely with hardware-backed protection.
- **BiometricAuthManager**: Controls authentication using fingerprint or face recognition.
- **Integrity Verification**: Uses SHA-256 hashing to ensure file authenticity.

4. Networking Layer (P2P Communication)

This layer manages device-to-device communication using WiFi Direct and Java sockets.

- Establishes peer discovery and connection.
- Handles secure handshake (encrypted session key exchange).
- Manages continuous encrypted data streaming between sender and receiver.

5. Data Layer (Persistence Layer)

This layer is responsible for storing and managing application data.

- **SQLite Database**: Stores transfer history, device information, and logs.
- **DAO (Data Access Objects)**: Provides structured access to database operations.
- Supports filtering, searching, and retrieval of past transfer records.

6. Utility Layer

This layer provides supporting functionalities required across the system:

- File handling (read/write operations)
- Compression (GZIP for optimized transfer)
- Secure file deletion (DoD 5220.22-M standard)

Architecture Flow

1. User initiates file transfer from the UI.
2. Application layer processes the request and triggers device pairing if needed.
3. Security layer generates and encrypts the session key.
4. Networking layer establishes a WiFi Direct connection and performs a secure handshake.
5. Encrypted file data is transmitted through sockets.
6. Receiver decrypts the file and verifies integrity.
7. Data layer logs the transfer details in SQLite.

4

4. Proposed Solution

The proposed system is an Android-based Secure File Share application that enables fast, reliable, and completely private file transfer using peer-topeer (P2P) communication. Unlike traditional file sharing methods that depend on internet connectivity or cloud servers, this solution uses WiFi Direct to establish a direct connection between devices, ensuring high-speed data transfer with zero reliance on external infrastructure.

The core of the proposed solution focuses on end-to-end security and user privacy. Each file transfer is protected using AES-256-GCM encryption, while RSA-2048 is used for securely exchanging session keys between devices. The use of Android Keystore ensures that private keys are stored securely with hardware-level protection, preventing unauthorized access or extraction. Additionally, biometric authentication (fingerprint or face unlock) adds an extra layer of security to restrict app access.

To simplify the connection process, the system introduces QR codebased device pairing, allowing users to quickly establish trust between devices without manual configuration. Once paired, devices can communicate securely, and trusted devices can be saved for future transfers, improving usability.

The application also enhances performance and user experience through features like batch file transfer, real-time progress tracking, and smart compression for efficient data transmission. After receiving files, the system verifies integrity using SHA-256 hashing to ensure that data is not altered or corrupted during transfer. Furthermore, a secure deletion mechanism based on DoD standards ensures that sensitive files cannot be recovered once deleted.

All transfer activities are recorded in a local SQLite database, enabling users to view history, filter records, and manage previously shared files. This makes the system not only secure but also organized and user-friendly.

Overall, the proposed solution provides a secure, offline, fast, and user-centric file sharing system, addressing the limitations of existing solutions by combining strong encryption, decentralized communication, and seamless usability into a single Android application.

5. Conclusion

The Secure File Share system provides a practical and secure solution for modern file transfer needs by eliminating dependency on internet connectivity and centralized servers. By leveraging WiFi Direct technology, the application enables fast and reliable peer-to-peer communication, ensuring efficient data transfer between devices.

The integration of advanced security mechanisms such as AES-256GCM encryption, RSA-2048 key exchange, and Android Keystore ensures strong protection of sensitive data during transmission and storage. Additional features like biometric authentication and SHA-256 integrity verification further enhance system security, making it resistant to unauthorized access and data tampering.

The system also focuses on usability by incorporating QR codebased pairing, batch file transfer, real-time progress tracking, and transfer history management using SQLite. These features make the application easy to use while maintaining high performance and reliability.

In conclusion, the proposed system successfully combines security, speed, and usability to create an efficient offline file sharing solution. It addresses the limitations of existing methods and provides a scalable foundation for future enhancements such as transfer resumption and multi-device connectivity.

6. Refferences

Android Developers, “Android Keystore System,” Google, 2024. Available: https://developer.android.com/privacy-and-security/keystore

J. Blessing, R. J. Anderson, and A. R. Beresford, “KeyDroid: A Large-Scale Analysis of Secure Key Storage in Android Apps,” University of Cambridge, 2025.

A. Pliekhov et al., “Wi-Fi Direct in Android: Creating Seamless Device-to-Device Communication,” 2025.

S. Tueno et al., “Pear2Pear (On WiFi): A Data Sharing Protocol Over WiFi through a Peer-to-Peer Network,” arXiv, 2019.

C. Casetti et al., “Content-Centric Routing in Wi-Fi Direct Multigroup Networks,” arXiv, 2014.

A. Verma, “Encryption and Real Time Decryption for Protecting Android Applications,” arXiv, 2021.

T. Burton, “Secure Implementation of AES-256 Encryption for Data Communication,” University of Oxford, 2019.

ITNEXT, “Hybrid Encryption in Android: Secure Communication Between Mobile Systems,” 2025.

International Journal of Research and Innovation, “Comparative Study of File Sharing Applications Using Wi-Fi Direct,” 2021.

Android Developers, “KeyProperties API Reference,” Google, 2026.