nternational Journal of Engineering and Techniques - Volume 12 Issue 2, March-April - 2026

RESEARCH ARTICLE OPEN ACCESS

Secure Peer to Peer File Sharing Application With End to End Encryption

Mohamed Imran M[1], Metildamary S[2], Mitheela K[3],Suvasthika N[4], Nidhya K[5]

[1]*Department of Computer Science and Engineering,Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu – 613006, India*
Email: imd048549@gmail.com

[2]*Department of Computer Science and Engineering,Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu – 613006, India*
Email: metisagayam@gmail.com

[3]*Department of Computer Science and Engineering,Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu 613006, India*
Email: mitheela.kr05@gmail.com

[4]*Department of Computer Science and Engineering,Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu – 613006, India*
Email: suvasthikanatarajan@gmail.com

[5]*Department of Computer Science and Engineering,Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu – 613006, India*
Email: nithya.marusihaa@gmail.com

Abstract:

This project presents a Secure Peer-to-Peer File Sharing Application with End-to-End Encryption (E2EE) to ensure data privacy and security during transmission. Traditional file sharing systems are vulnerable to data breaches, unauthorized access, and dependency on centralized servers. The proposed system overcomes these issues by using a decentralized peer-to-peer architecture. It implements Advanced Encryption Standard (AES) for data encryption and RSA algorithm for secure key exchange, ensuring that only authorized users can access the shared files. End-to-end encryption protects data from interception and tampering. The system also includes secure key management and user authentication to enhance security. It supports efficient real-time file transfer with minimal delay. Overall, the proposed solution provides a secure, reliable, and scalable approach for modern file sharing needs. Keywords—**Peer-to-Peer (P2P), End-to-End Encryption (E2EE), AES, RSA, Secure File Sharing, Data Security.**

I. INTRODUCTION

Secure file sharing has become a critical requirement in modern digital communication due to the rapid growth of internet usage and the increasing need for data exchange across networks. Traditional file sharing systems primarily depend on centralized servers, which are often vulnerable to various security threats such as data breaches, unauthorized access, data tampering, and single points of failure. These limitations raise serious concerns regarding data privacy, confidentiality, and reliability, especially in sensitive applications. With the rise in cyber attacks

and privacy issues, there is a growing demand for secure and efficient file sharing solutions.

Peer-to-peer (P2P) architecture offers a decentralized approach where users can communicate and share files directly without relying on a central authority. This reduces dependency on servers and improves system scalability and availability. However, ensuring secure communication in P2P networks remains a significant challenge due to risks such as interception, man-in-the-middle attacks, and unauthorized data access. To overcome these challenges, strong encryption and secure key management techniques are required.

This project proposes a Secure Peer-to-Peer File Sharing Application with End-to-End Encryption (E2EE) to enhance data security and user privacy. The system uses Advanced Encryption Standard (AES) for encrypting file data and RSA algorithm for secure key exchange between users. End-to-end encryption ensures that the data remains encrypted throughout the transmission process and can only be accessed by the intended receiver. In addition, the system incorporates user authentication and secure session management to prevent unauthorized access.

The proposed system also supports efficient and real-time file transfer while maintaining high security standards. By integrating decentralized communication, robust encryption mechanisms, and secure key management, this application provides a reliable, scalable, and secure solution for modern file sharing needs. This approach significantly improves data protection and ensures safe communication in today's digital environment.

II. RELATED WORK

Several existing studies and applications have focused on secure file sharing and data protection in distributed systems. Traditional cloud-based file sharing platforms such as Google Drive and Dropbox provide convenient storage and sharing features but rely heavily on centralized servers, which can lead to privacy concerns, data breaches, and lack of user control over data. To overcome these limitations, researchers have explored peer-to-peer (P2P) based file sharing systems that eliminate the need for central servers and improve scalability and availability.

Earlier P2P systems like BitTorrent enable efficient file distribution but lack strong security mechanisms, making them vulnerable to data interception and unauthorized access. To address security challenges, various encryption-based approaches have been proposed. End-to-end encryption (E2EE) has gained significant attention as it ensures that data is encrypted at the sender side and can only be decrypted by the intended receiver, preventing third-party access. Several secure communication applications such as WhatsApp and Signal successfully implement E2EE to protect user data during transmission.

In addition, many research works have utilized cryptographic algorithms such as Advanced Encryption Standard (AES) for data encryption And RSA for secure key exchange to enhance security in file sharing systems. Some systems also incorporate authentication mechanisms and secure key management techniques to prevent attacks such as man-in-the-middle and unauthorized access. However, many of these solutions either compromise performance, depend on partial centralization, or lack efficient real-time file transfer capabilities.

The proposed system builds upon these existing approaches by combining decentralized P2P architecture with strong encryption techniques, secure key management, and efficient file transfer mechanisms. This integration aims to provide a more secure, reliable, and scalable file sharing solution compared to existing systems.

III. PROPOSED METHODOLOGY

The proposed system is designed as a Secure Peer-to-Peer File Sharing Application with End-to-End Encryption (E2EE) to ensure secure and reliable data transmission between users. The system follows a decentralized architecture where files are shared directly between sender and receiver without relying on a centralized server. Initially, users register and authenticate themselves using secure login credentials to prevent unauthorized access. Once authenticated, a secure session is established between peers.

When a user initiates file sharing, the file is first encrypted using the Advanced Encryption Standard (AES) algorithm to ensure data confidentiality. A unique symmetric key is generated for each file transfer. This AES key is then

securely exchanged using the RSA algorithm, where the receiver's public key is used to encrypt the symmetric key. Only the intended receiver, possessing the corresponding private key, can decrypt and access the symmetric key.

After successful key exchange, the encrypted file is transmitted over the network through a direct peer-to-peer connection. End-to-end encryption ensures that the file remains protected throughout the transmission process and cannot be accessed by any third party or intermediary. On the receiver side, the encrypted file is decrypted using the received AES key, restoring the original data.

The system also incorporates secure key management techniques to dynamically generate and handle encryption keys, reducing the risk of key leakage. Additionally, mechanisms such as session validation and secure communication protocols are implemented to prevent attacks like man-in-the-middle and data tampering. The overall methodology ensures a balance between security, performance, and scalability, making the system suitable for modern secure file sharing applications.

IV. SYSTEM ARCHITECTURE

The proposed system follows a secure peer-to-peer architecture designed to enable direct and encrypted file transfer between sender and receiver within a LAN environment. The architecture consists of three main components: Sender Application, Network Communication Layer, and Receiver Application. The Sender Application includes modules such as User Interface (UI), Authentication Module, ECDH Key Generation, File Selector, and AES-256-GCM Encryption Module. The process begins with user authentication, followed by file selection for transmission. The system then generates cryptographic keys using Elliptic Curve Diffie-Hellman (ECDH) to establish a shared secret key between sender and receiver.

The Network Communication Layer facilitates secure data transfer using UDP-based discovery service to identify available peers within the LAN network, followed by establishing a reliable TCP secure channel for communication. The shared secret derived using ECDH is further processed using HKDF (HMAC-based Key Derivation Function) to generate strong encryption keys. The selected file is encrypted using AES-256-GCM, which ensures both data confidentiality and integrity by producing ciphertext along with initialization vector (IV) and authentication tag.

The encrypted data is transmitted through the secure TCP channel to the Receiver Application. On the receiver side, the architecture includes modules such as UI Layer, Authentication Module, ECDH Key Generation, AES-256-GCM Decryption Module, and File Storage. The receiver uses its private key to derive the shared secret and regenerate the encryption key using HKDF. The received ciphertext is then decrypted using AES-256-GCM, verifying the integrity of the data using the authentication tag before storing the file securely.

This architecture ensures end-to-end encryption, secure key exchange, and efficient peer discovery while eliminating dependency on centralized servers. By combining ECDH, HKDF, AES-256-GCM, and secure TCP communication, the system provides a highly secure, reliable, and scalable solution for peer-to-peer file sharing.

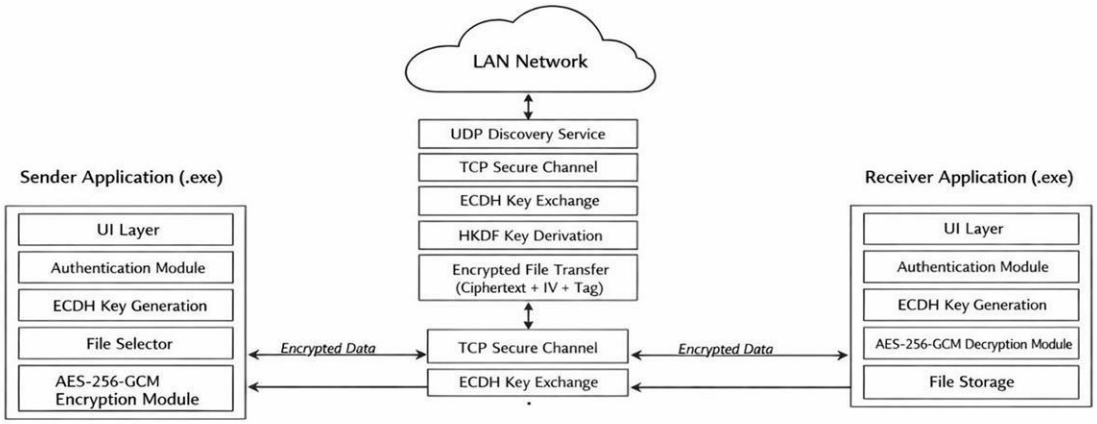

Fig 1: Secure Peer to Peer File Sharing Application With End to End Encryption

V. EXPERIMENTAL SETUP

The experimental setup of the proposed system is designed to evaluate the performance, security, and reliability of the secure peer-to-peer file sharing application in a controlled environment. The system is implemented and tested within a local area network (LAN) to simulate real-time

file transfer between sender and receiver. Various parameters such as encryption efficiency, key exchange process, data integrity, and transmission speed are analyzed to validate the effectiveness of the proposed approach.

a) *System Environment*— The application is developed as a desktop-based executable (.exe) using suitable programming tools and cryptographic libraries. The testing is carried out on multiple systems connected through a LAN network to enable direct peer-to-peer communication. The setup uses standard hardware configurations to ensure that the system performs efficiently in real-world scenarios.

b) *Peer Discovery Mechanism* — The system utilizes a UDP-based discovery service to identify available peers within the local network. This mechanism allows sender and receiver to detect each other dynamically without manual configuration. The performance of peer discovery is evaluated based on speed and accuracy in identifying active nodes.

c) *Secure Key Exchange* — The application implements Elliptic Curve Diffie-Hellman (ECDH) for establishing a shared secret key between peers. Both sender and receiver generate their respective key pairs and exchange public keys securely. The shared key is then derived using HKDF to produce a strong encryption key, ensuring secure communication.

d) *Encryption and Decryption Process* — The file selected for transfer is encrypted using AES-256-GCM algorithm, which provides both confidentiality and integrity. The encrypted output includes ciphertext, initialization vector (IV), and authentication tag. At the receiver side, the same key is used to decrypt the file and verify its integrity before storing it.

e) *Data Transmission* — The encrypted data is transmitted through a reliable TCP secure channel established between the sender and receiver. The experiment measures the efficiency of data transfer, packet delivery, and consistency of communication during real-time file sharing.

f) *Performance and Security Evaluation* — The system is analyzed based on metrics such as encryption time, decryption time, latency, and file transfer speed. Security evaluation ensures that the system is resistant to threats such as data interception, unauthorized access, and tampering. The results confirm that the system maintains high security and efficient performance.

VI. CONCLUSIONS

The proposed system successfully implements a secure peer-to-peer file sharing application that ensures safe and reliable transmission of files between sender and receiver. The system utilizes advanced cryptographic techniques such as Elliptic Curve Diffie-Hellman (ECDH) for secure key exchange, HKDF for key derivation, and AES-256-GCM for encryption, providing strong data confidentiality and integrity. The encryption mechanism ensures that file contents remain protected throughout transmission, while authentication tags help detect any tampering attempts. The use of TCP protocol guarantees reliable and ordered data transfer between communicating devices. Additionally, the generation of temporary session keys enhances security by protecting each communication session from unauthorized access. Overall, the proposed system demonstrates an effective and practical approach for securing data during file sharing and highlights the importance of integrating cryptographic techniques in modern secure communication systems.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their project guide and faculty members of the Department of Computer Science and Engineering for their continuous support, valuable guidance, and encouragement throughout this research work.

The authors also thank their external guide associated with Cyber Nerds for providing the necessary guidance and resources to successfully complete this study.

Additionally, we acknowledge the use of publicly available datasets and tools that contributed to the development and evaluation of the proposed model.

REFERENCES

1. *Diffie and M. Hellman, "New Directions in Cryptography,"* IEEE Transactions on *Information Theory, vol. 22, no. 6, pp. 644–654, 1976.*

2. *R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.*

3. *National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," FIPS PUB 197, 2001.*

4. *H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)," RFC 5869, 2010.*

5. *V. S. Miller, "Use of Elliptic Curves in Cryptography," Advances in Cryptology—CRYPTO, 1985.*

6. *D. J. Bernstein, "Curve25519: New Diffie-Hellman Speed Records," Public Key Cryptography, 2006.*

7. *B. Cohen, "Incentives Build Robustness in BitTorrent," Workshop on Economics of Peer-to-Peer Systems, 2003.*

8. *A. Langley et al., "The QUIC Transport Protocol: Design and Internet-Scale Deployment," ACM SIGCOMM, 2017.*

9. *M. Green and M. Smith, "The Cryptopals Crypto Challenges," Cryptography Engineering, 2016.*

10. *K. P. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," NIST Special Publication, 2007.*

11. *C. Paar and J. Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, Springer, 2010.*

12. *W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed., Pearson, 2017.*