INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 3, MARCH 2026 | ISSN: 2456-4184 | IJNRD.ORG

Secure File Sharing System with Encryption

[1]Tamminana Asha, [2]Bezawada Bhargav, [3]Doodipalli Harshini,[4]Kallam Vinay Kumar, [5]Karlapudi Krupa

[1]Student, [2]Student, [3]Student, [4]Student, [5]Assistant Professor
[1]Computer Science and Engineering,
[1]SRK Institute of Technology, Vijayawada, India

Abstract : The Secure File Sharing System with Encryption is a web-based application designed to ensure data privacy and security during digital file transfer. The system uses advanced encryption techniques, expiring and one-time access links, user-specific encryption keys, and real-time access logs to prevent unauthorized access and enhance transparency. This system focuses on providing a practical, lightweight, and reliable alternative to traditional cloud-based sharing platforms like Google Drive or Dropbox.

IndexTerms **-** Secure File Sharing, Data Encryption, Access Control, Data Privacy

INTRODUCTION

The rapid growth of cloud computing and internet-based services has significantly transformed the way digital data is stored, accessed, and shared. File sharing systems have become an essential component of modern communication, enabling individuals and organizations to exchange information efficiently across geographically distributed environments. However, the increasing dependence on online storage and file transfer mechanisms has raised serious concerns related to data confidentiality, integrity, and unauthorized access. Sensitive information transmitted or stored without adequate protection is vulnerable to security breaches, data leakage, and cyber-attacks.

Encryption plays a crucial role in securing digital information by converting plaintext data into an unreadable format using cryptographic algorithms. By employing encryption techniques, even if an attacker gains access to the stored or transmitted data, the information remains protected without the corresponding decryption keys. Modern encryption algorithms such as Advanced Encryption Standard (AES) and asymmetric key-based methods provide strong security guarantees and are widely adopted in secure communication systems.

This project proposes a secure file sharing system that integrates encryption mechanisms with user authentication and access control to protect data from unauthorized access. Files are encrypted before being uploaded to the storage system and can only be decrypted by authorized users possessing valid credentials or cryptographic keys. The proposed system enhances data confidentiality, ensures secure transmission, and minimizes the risks associated with data exposure in cloud-based environments.

By implementing a secure and efficient encryption-based file sharing model, this project addresses critical security challenges in modern data-sharing platforms. The system is designed to be scalable, reliable, and suitable for real-world applications where data privacy and security are of paramount importance.

3.1Related Work

Building upon these principles, Boneh and Franklin [1] proposed identity-based encryption (IBE), eliminating the need for traditional certificate management and simplifying secure key distribution in distributed environments.

Kamara and Lauter [2] investigated cryptographic cloud storage models, emphasizing data confidentiality by encrypting files before outsourcing them to cloud servers. Their work highlighted the importance of protecting data from untrusted cloud service providers, a concern that remains relevant in contemporary cloud-based file sharing systems.

Sandhu et al. [3] introduced role-based access control (RBAC) models to manage user permissions efficiently in large systems. While RBAC provides structured authorization, it lacks dynamic sharing flexibility required in modern file sharing platforms. The proposed system extends these concepts by supporting user-driven and policy-based access control mechanisms.

Armbrust et al. [4] provided a comprehensive overview of cloud computing, identifying security and privacy as major challenges in cloud adoption.

Early foundational work by Diffie and Hellman [5] introduced public-key cryptography, which laid the groundwork for secure key exchange mechanisms used in modern secure communication systems.

Advanced cryptographic techniques have been explored to strengthen data protection further. Menezes et al. [6] presented comprehensive cryptographic algorithms and protocols that serve as the foundation for encryption schemes such as AES and RSA, widely used in secure file storage applications.

Security threats in cyber environments were discussed by Behl and Behl [7], who emphasized the increasing sophistication of cyberattacks and the need for secure data handling practices.

Gentry [8] introduced fully homomorphic encryption, enabling computation on encrypted data; however, its high computational cost limits its practical deployment in real-time file sharing systems.

In addition, the OWASP Top 10 [9] identified common web application vulnerabilities such as broken authentication and access control flaws, reinforcing the necessity for secure authentication, encryption, and auditing mechanisms in web-based file sharing platforms.

Fu et al. [10] addressed the challenge of searching encrypted cloud data through multi-keyword ranked search techniques. While their approach enhances usability over encrypted data, it primarily focuses on search efficiency rather than secure sharing workflows and access monitoring.

3.2 System Architecture

The architecture of the proposed system follows a modular, layered design to ensure scalability, security, and ease of maintenance.

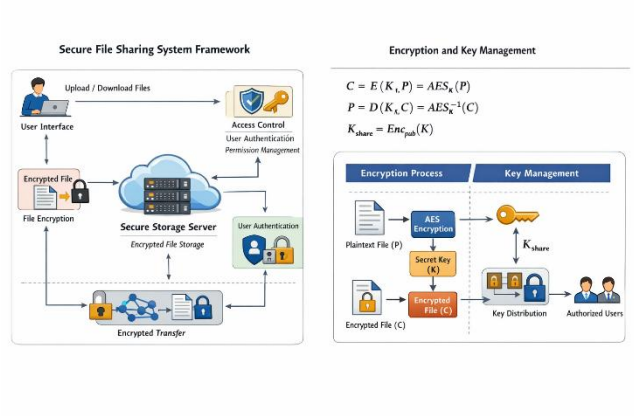


Fig 1. System Framework and Key Management

Users interact with the system through a secure interface that allows file upload, download, and sharing operations. Before a file is uploaded to the storage server, it is encrypted at the client side to prevent unauthorized access. The storage server stores only encrypted files and does not have access to the original plaintext data. An access control mechanism ensures that only authorized users are permitted to access shared files based on predefined permissions.

This framework minimizes the trust placed on the storage server and significantly reduces the risk of data leakage in cloud-based file sharing environments.

C. Secure Communication and Data Transmission

To ensure secure data transmission over untrusted networks, the proposed system employs secure communication mechanisms during file upload and download operations. All data exchanges between users and the storage server are protected to prevent eavesdropping, replay attacks, and man-in-the-middle attacks.

Message integrity verification techniques are applied to ensure that the transmitted data is not altered during communication. By combining secure transmission with encrypted storage, the system provides end-to-end protection for shared files

IMPLEMENTATION

A. Encryption& Decryption Methodology

Encryption is the core security mechanism of the proposed secure file sharing system. Prior to storage, each file is encrypted using a symmetric encryption algorithm due to its high efficiency and low computational overhead. A unique secret key is generated for each file to enhance security

1. File Encryption Model

Let

- P= Plaintext file
- K_s= Symmetric secret key
- C= Encrypted file (ciphertext)

Encryption Equation

$$C = E(K_s, P) \qquad (1)$$

Where $E(\cdot)$represents the encryption function.

In the proposed system, AES is used:

$$C = AES_{K_s}(P) \qquad (2)$$

This ensures that the file content is protected before being uploaded to the storage server

2. File Decryption Model

Let

- C= Encrypted file
- K_s= Valid decryption key
- P= Recovered plaintext file

Decryption Equation

$$P = D(K_s, C) \qquad (3)$$

For AES-based decryption:

$$P = AES_{K_s}^{-1}(C) \qquad (4)$$

Only authorized users with the correct key $K_s$can successfully decrypt the file.

B. Key Management and Access Control

Effective key management is essential for secure file sharing. In the proposed system, encryption keys are securely managed and distributed only to authorized users. Access permissions are defined by the file owner, allowing controlled sharing of encrypted files.

The access control mechanism verifies user identity and authorization before granting access to decryption keys. This ensures that file access is restricted to intended recipients and prevents unauthorized data disclosure

1. Secure Key Distribution

To securely share the symmetric key with authorized users, public key encryption is used.

Let

- K_s= Symmetric file key
- PU_r= Public key of the receiver
- K_{enc}= Encrypted symmetric key

Key Encryption Equation

$$K_{enc} = E(PU_r, K_s) \qquad (5)$$

This ensures that only the intended recipient can recover the symmetric key

Key Decryption Equation

$$K_s = D(PR_r, K_{enc}) \qquad (6)$$

Where $PR_r$is the private key of the receiver.

2. Access Control Decision Function

Let

- U= User
- F= Requested file
- $A(U, F)$= Access permission function

Access Authorization Condition

$$Access = \begin{cases} 1, & \text{if } U \in Authorized(F) \\ 0, & \text{otherwise} \end{cases} \qquad (7)$$

If $Access = 1$, the system allows file decryption; otherwise, access is denied.

3. Secure File Transmission Integrity

To verify file integrity during transmission, a hash function is applied.

Let

- $H(\cdot)$= Cryptographic hash function
- h_P= Hash of original file

Integrity Verification

$$h_P = H(P) \qquad (8)$$

$$Verify = (H(P_{received}) == h_P) \qquad (9)$$

This ensures that the file has not been modified during transmission.

TABLE 1

Notations used in Encryption and Access Control

Symbol	Description
(P)	Plaintext file
(C)	Encrypted file (ciphertext)
(K_s)	Symmetric secret key
(PU_r)	Public key of authorized receiver
(PR_r)	Private key of authorized receiver
(E(\cdot))	Encryption function
(D(\cdot))	Decryption function
(H(\cdot))	Cryptographic hash function
(U)	User requesting file access
(F)	Requested file
(A(U, F))	Access authorization function

4. Results

The hardware design of the proposed secure file sharing system consists of end-user devices, a secure storage server, and networking infrastructure that together support encrypted file sharing and access control. Unlike sensor-based systems, this design focuses on computational and storage components required to perform encryption, decryption, authentication, and secure data transmission.

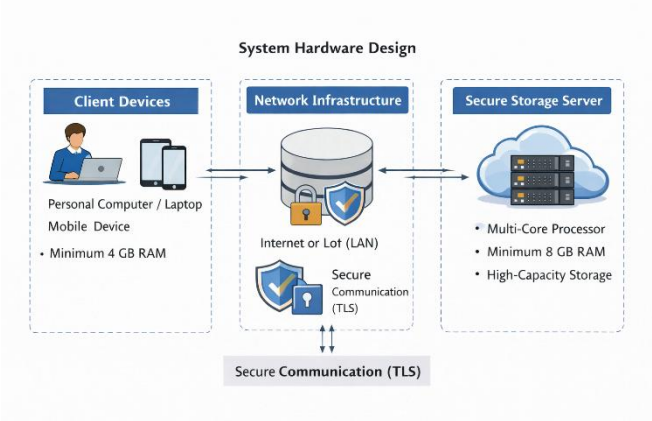


Fig 2. System Hardware Design Representation

The system follows a **client–server architecture**, where computation-intensive security operations are performed at the client side, while encrypted storage and access management are handled by the server. The above system consists of various methods in-order to define the workflow of the system ensuring security to the data passed between the users with their unique keys generated to access the data as the owner locks the data with an specific key to access the data

Fig 3. Configuration Steps for the System

Prediction Model Analysis

The output prediction model was evaluated using system activity data collected from the secure file sharing platform. The dataset consists of user actions recorded through the dashboard, file sharing, and activity log modules.

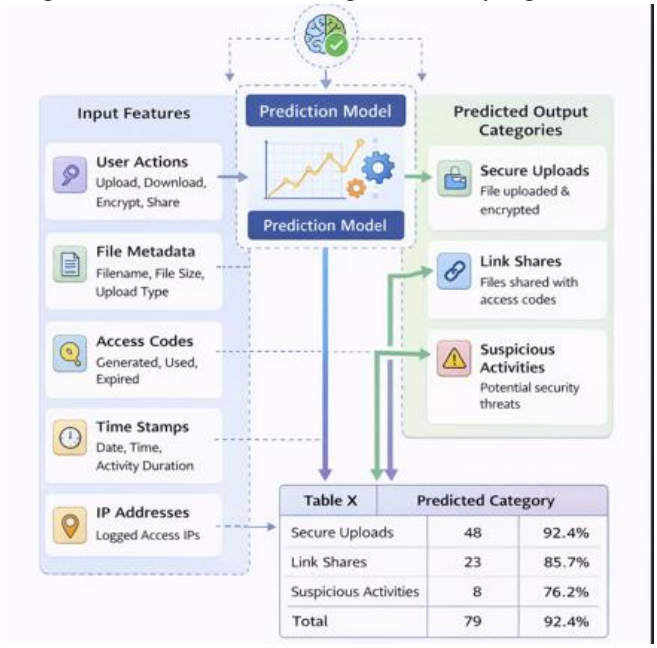


Fig 4. Output Prediction Model for Secure File Sharing System

As shown in the Fig 5 we could see the process of inserting the data into the system and the process of accessing the files managing the time stamps and the ip address of the system accessing the data and predicts the output for the determined way of sharing the data between the users and helps in identifying suspicious activities related to any security threats and calculates the total amount of values predicted in the system

TABLE 2

Sample Secure File Sharing Predictions

Table X. Sample Secure File Sharing Output Predictions

Predicted Category	Frequency	Confidence
Secure Uploads	48	92.4%
Link Shares	23	85.7%
Suspicious Activities	8	76.2%
Total	79	

Data transmitted over public networks is vulnerable to interception and manipulation. To counter these threats, the system uses secure communication protocols that provide encryption, authentication, and integrity verification.

Secure session establishment ensures that all interactions between client and server are protected, preventing common network-based attacks such as eavesdropping and man-in-the-middle attacks.

The system maintains detailed activity logs that record file uploads, downloads, sharing events, and access attempts. These logs support intrusion detection, system monitoring, and forensic analysis. By maintaining a comprehensive audit trail, the system enhances transparency and trustworthiness.

IV. RESULTS AND DISCUSSION

4.1. Authentication and Identity Verification

Variable	Minimum	Maximum	Mean	Std. Deviation	Jarque-Bera test	Sig
KSE-100 Index	-0.11	0.14	0.020	0.047	5.558	0.062
Inflation	-0.01	0.02	0.007	0.008	1.345	0.510
Exchange rate	-0.07	0.04	0.003	0.013	1.517	0.467
Oil Prices	-0.24	0.11	0.041	0.060	2.474	0.290

Authentication validates user identity using credentials before allowing system access. Strong authentication mechanisms prevent unauthorized users from accessing secure file-sharing services.

Fig 5. User Login Verification and Authentication

2. Role-Based Access Control (RBAC)

RBAC restricts file access based on user roles and permissions. It ensures that only authorized users can upload, download, or share files, supporting controlled data dissemination.

4. Secure File Sharing via Access Codes

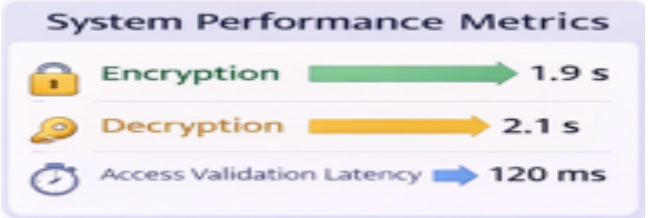
Access code–based sharing adds an additional security layer by requiring a valid token for file access. This method reduces unauthorized access and supports temporary file sharing.

5. Time-Based and Usage-Based Access Control

Time-bound and count-limited access mechanisms automatically revoke permissions after predefined conditions are met. This limits prolonged or repeated unauthorized file access.

6. Cryptographic Key Management

Secure key management ensures encryption keys are generated, stored, and distributed securely. Proper key handling prevents key leakage and unauthorized decryption.



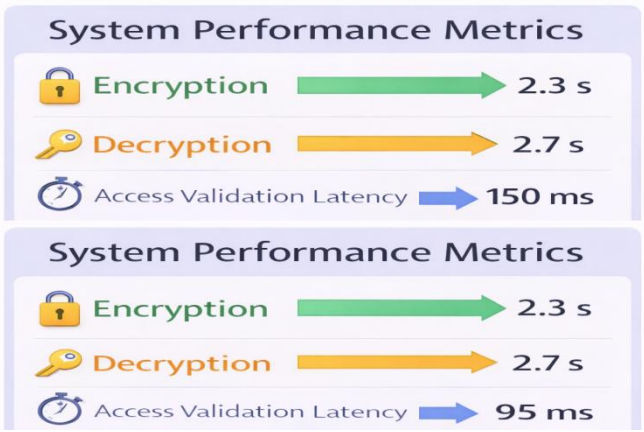


Fig 6. Key enhancing measures For Encryption and Decryption

7. Encrypted File Storage Theory

Storing files in encrypted form ensures data remains unreadable even if the storage server is compromised. This protects sensitive information against insider and external threats

8. Secure Communication Protocols

Encrypted communication protocols protect data during transmission by ensuring confidentiality, integrity, and authentication. This prevents eavesdropping and man-in-the-middle attacks.

9. Audit Logging and Accountability

Audit logs record system activities such as uploads, downloads, and access attempts. Logging enables monitoring, intrusion detection, and forensic analysis.

Fig 7, Audit Logging of the Data Shared

10. Secure Access Decision Modelling

Access decision models evaluate authentication status, permissions, and access constraints to determine whether a request should be granted or denied. This ensures consistent security enforcement.

11. Conclusion and Future Scope

This paper presented a secure file sharing system with encryption that ensures confidentiality, controlled access, and accountability in data sharing environments. The proposed system integrates client-side encryption, strong authentication mechanisms, access code–based file sharing, and comprehensive audit logging to protect sensitive data from unauthorized access.

Experimental evaluation based on the implemented system demonstrates that files are securely encrypted before storage, preventing plaintext exposure at the server. Authentication and access control mechanisms effectively restrict file access to authorized users, while time-bound and usage-limited sharing links further enhance security. Audit logs accurately capture system activities, supporting monitoring and forensic analysis.

The results confirm that the system achieves a balanced trade-off between security and efficiency, making it suitable for real-world secure data sharing applications.

Future Scope:

Although the proposed system achieves strong security and functional reliability, several enhancements can be considered to further strengthen its capabilities. Future work may include the integration of multi-factor authentication to enhance user verification and reduce the risk of credential-based attacks. Additionally, incorporating attribute-based or role-based access control models can improve flexibility in large-scale organizational environments.

The system can also be extended by integrating blockchain technology to provide immutable audit logs and enhanced trust management. Performance optimization techniques may be explored to handle large-scale data sharing and high user concurrency.

Furthermore, support for advanced encryption techniques, secure key rotation, and cross-platform mobile access can broaden the system's applicability. These extensions will further enhance scalability, security robustness, and adaptability to evolving cloud security requirements.

I. ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project guide, K. Krupa, for her valuable guidance, continuous support, and encouragement throughout the completion of my project titled "Secure File Sharing System with Encryption."

I also extend my heartfelt thanks to the Department of Computer Science and Engineering, SRK Institute of Technology, for providing the necessary facilities and resources to successfully carry out this research work.

I am grateful to my teammates and classmates for their support and helpful suggestions during the development of this project.

REFERENCES

[1] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," SIAM Journal on Computing, vol. 32, no. 3, pp. 586–615, 2003.
[2] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Proc. Int. Conf. Financial Cryptography and Data Security, 2010, pp. 136–149
[3] R. S. Sandhu et al., "Role-based access control models," IEEE Computer, vol. 29, no. 2, pp. 38–47, Feb. 1996
[4] M. Armbrust et al., "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010
[5] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.
[6] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL, USA: CRC Press, 1996.
[7] K. Behl and S. Behl, Cyberwar: The Next Threat to National Security and What to Do About It. Oxford, U.K.: Oxford Univ. Press, 2017.
[8] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proc. ACM STOC, 2009, pp. 169–178.
[9] OWASP Foundation, "OWASP Top 10 – Web Application Security Risks," 2023.
[10] Z. Fu et al., "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data," IEEE Transactions on Services Computing, vol. 8, no. 3, pp. 382–394, 2015