DEVELOPMENT OF A SECURE FILE STORAGE SYSTEM USING HYBRID ENCRYPTION TECHNIQUES

[1]Ameh, Friday [2] Asogwa T.C. (PhD). [1,2] Department of Computer Science Enugu State University of Science and Technology, Agbani, Enugu State, Nigeria
Email: friday4ict@gmail.com

ABSTRACT

The rapid growth of digital data and cloud computing has intensified concerns regarding data confidentiality, integrity, and secure access control. Traditional file storage systems relying solely on symmetric or asymmetric encryption often experience trade-offs between performance efficiency and secure key management. This study presents the design and implementation of a Secure File Storage System (SFSS) based on hybrid encryption techniques that integrate symmetric encryption (AES) and asymmetric encryption (RSA) to enhance data protection. In the proposed system, files are encrypted using a randomly generated Advanced Encryption Standard (AES) key for efficiency, while the AES key is encrypted using the Rivest–Shamir–Adleman (RSA) public key for secure key distribution. A secure key management mechanism and integrity verification using SHA-256 hashing are incorporated to prevent unauthorized modification. The system was implemented using the Laravel PHP framework and deployed on a cloud-based platform. Performance evaluation demonstrates that the hybrid encryption model achieves strong security guarantees while maintaining acceptable encryption and decryption latency. The findings confirm that hybrid encryption provides improved confidentiality, integrity, and scalability for secure file storage applications in enterprise and cloud environments.

Keywords: Hybrid Encryption, Secure File Storage, AES, RSA, Cloud Security, Key Management, Data Integrity

1. INTRODUCTION

1.1 Background of the Study

The proliferation of cloud computing and digital transformation has significantly increased the volume of data stored electronically. Organizations now rely heavily on cloud-based storage solutions for scalability and accessibility. However, this dependence exposes sensitive information to cyber threats such as data breaches, ransomware attacks, insider threats, and unauthorized access. With the technology developed in today's era, data growth has raised serious challenges when it comes to secure storage, especially sensitive data. Data storage mechanisms should ensure not just data integrity and access but confidentiality of information stored. Solutions for ensuring enhanced security are available through the use of hybrid encryption, a powerful method. Conventional encryption mechanisms typically rely on either

symmetric or asymmetric cryptography. Symmetric encryption algorithms such as AES provide high-speed data encryption but present challenges in secure key distribution. Conversely, asymmetric encryption algorithms such as RSA offer secure key exchange mechanisms but are computationally expensive for large data volumes. Hybrid encryption integrates both techniques, leveraging the strengths of each while mitigating their weaknesses. Hybrid encryption benefits from symmetric and asymmetric encryption mechanisms' strengths for securing data. This paper introduces a hybrid encryption-based model for safe file storage, investigating its design and implementation. Hybrid encryption combines the efficiency of symmetric encryption with the security of asymmetric encryption. In this model, a symmetric key encrypts the actual data since it is efficient, while asymmetric encryption is employed to protect the symmetric key to facilitate secure key exchange and enhanced security measures. The dual-layered approach addresses the disadvantage of using one encryption algorithm alone, thus achieving a secure scheme for file storage systems (Chen et al., 2019). The significance of employing hybrid encryption in file storage solutions is that it can maximize performance without compromising the level of security. As cyber-attacks and data breaches increase, organizations have no option but to accord high priority to protecting sensitive data from unauthorized access (Ranjan & Kumar, 2020). This study proposes a hybrid encryption-based secure file storage system that enhances confidentiality, integrity, and secure key management in cloud environments.

2. RELATED WORK

The security of file storage systems has received significant scholarly attention, particularly with the rapid expansion of cloud computing and distributed data environments. Early approaches primarily relied on symmetric encryption algorithms such as the Advanced Encryption Standard (AES) due to their computational efficiency and suitability for large data volumes. AES remains widely adopted because of its strong security guarantees and high throughput performance (Stallings, 2021; Shay et al., 2021). However, symmetric encryption presents notable challenges in secure key distribution and management, especially in multi-user and cloud-based systems (Katz & Lindell, 2020).

To address these limitations, researchers introduced asymmetric cryptographic techniques such as the Rivest–Shamir–Adleman (RSA) algorithm for secure key exchange. Asymmetric encryption enhances authentication and key protection mechanisms but incurs substantial

computational overhead when applied directly to bulk data encryption (Boneh & Venkatesan, 2019; Albrecht & Renard, 2021). Consequently, purely asymmetric encryption models are often inefficient for secure file storage applications.

Hybrid encryption frameworks emerged as an effective solution by integrating the efficiency of symmetric encryption with the secure key management capabilities of asymmetric cryptography. In such systems, data are encrypted using AES, while the symmetric key is secured using RSA (Bellare & Rogaway, 2021). Empirical studies demonstrate that hybrid encryption significantly improves both security and performance in cloud storage environments (Li et al., 2020; Zafar & Aziz, 2020). Nevertheless, some implementations lack comprehensive integrity verification mechanisms and robust key lifecycle management.

Recent studies emphasize the importance of integrating hashing algorithms such as SHA-256 for data integrity verification and implementing structured key management frameworks to reduce vulnerability risks (Kumar & Jha, 2022; Wang et al., 2021). Despite these advancements, gaps remain in scalable deployment, user-friendly key management, and empirical performance benchmarking of hybrid encryption-based storage systems.

Therefore, the present study builds upon existing research by implementing a deployable secure file storage system that integrates AES-based data encryption, RSA-based key protection, SHA-256 integrity verification, and structured key management within a cloud-compatible web framework. This approach addresses limitations identified in prior studies and contributes a practical and scalable hybrid encryption model for secure file storage applications.

3. SYSTEM ARCHITECTURE

3.1 Hybrid Encryption Model

The proposed Secure File Storage System (SFSS) is designed using a layered hybrid encryption architecture that integrates confidentiality, integrity, and access control mechanisms within a unified framework. The architecture combines symmetric and asymmetric cryptographic techniques with secure key management and authentication controls to ensure robust data protection.

At the core of the system is the **Advanced Encryption Standard (AES)**, which performs symmetric encryption of uploaded files. AES is selected due to its high computational efficiency

and suitability for encrypting large data volumes with minimal latency. By encrypting file contents using a randomly generated session key, the system ensures strong confidentiality while maintaining performance scalability.

To address the challenge of secure key distribution, the symmetric AES key is encrypted using the **Rivest–Shamir–Adleman (RSA)** asymmetric algorithm. RSA enables secure transmission and storage of the AES key by encrypting it with the recipient's public key, ensuring that only the corresponding private key holder can decrypt it. This mechanism strengthens key protection and eliminates vulnerabilities associated with direct symmetric key sharing.

For data integrity assurance, the system incorporates **SHA-256 hashing**. A cryptographic hash of the encrypted file is generated and stored alongside it. During file retrieval, the hash is recomputed and compared with the stored value to verify that the file has not been altered. This guarantees tamper detection and preserves data integrity.

A structured **Key Management System (KMS)** is integrated to oversee key generation, secure storage, retrieval, and lifecycle management. The KMS ensures that cryptographic keys are protected against unauthorized access while supporting secure key rotation and controlled access policies.

Finally, an **Authentication Module** enforces role-based access control (RBAC), ensuring that only authorized users can upload, retrieve, or decrypt stored files. This module strengthens system security by binding cryptographic operations to verified user identities.

3.2 Encryption Process

The encryption process of the proposed Secure File Storage System (SFSS) follows a structured hybrid cryptographic workflow designed to ensure confidentiality, integrity, and secure key management (Bellare & Rogaway, 2021; Katz & Lindell, 2020). The process begins when a user uploads a file to the system through an authenticated interface. Upon receiving the file, the system automatically generates a unique, random Advanced Encryption Standard (AES) session key. The use of a session-based symmetric key enhances security by ensuring that each file is protected independently and reduces the risk associated with key reuse (Stallings, 2021).

The uploaded file is then encrypted using the generated AES key. Symmetric encryption is selected at this stage due to its high computational efficiency and suitability for encrypting large

data volumes (Shay et al., 2021). Following encryption, a Secure Hash Algorithm (SHA-256) digest of the encrypted file is computed. This cryptographic hash serves as an integrity verification mechanism, enabling the system to detect any unauthorized modification during storage or transmission (Wang et al., 2021). Hash-based integrity verification strengthens protection against tampering and replay attacks.

To secure the symmetric key, the system encrypts the AES key using the recipient's RSA public key. This asymmetric encryption step ensures that only the authorized user, possessing the corresponding RSA private key, can decrypt and access the symmetric key required for file decryption (Boneh & Venkatesan, 2019). By separating bulk data encryption from key protection, the hybrid encryption model combines computational efficiency with secure key exchange, thereby enhancing the overall security architecture (Li et al., 2020)

3.3 Decryption Process

The decryption phase of the Secure File Storage System (SFSS) is designed to securely restore encrypted data while ensuring confidentiality and integrity. The process begins when an authenticated user submits a file retrieval request through the system interface. Upon authorization, the system retrieves the encrypted file, the RSA-encrypted AES session key, and the stored SHA-256 hash value from cloud storage.

To reconstruct the original file, the system first applies the user's RSA private key to decrypt the encrypted AES session key. Asymmetric cryptography is employed at this stage to ensure that only the legitimate key owner can recover the symmetric key required for file decryption (Boneh & Venkatesan, 2019). This approach strengthens secure key exchange and prevents unauthorized access to encrypted content (Katz & Lindell, 2020).

Once the AES session key is successfully recovered, it is used to decrypt the encrypted file. Symmetric decryption is computationally efficient and well-suited for restoring large data volumes without introducing significant latency (Stallings, 2021). After decryption, the system recomputes the SHA-256 hash of the decrypted file and compares it with the previously stored hash value. Hash comparison serves as an integrity verification mechanism, ensuring that the file has not been tampered with during storage or transmission (Wang et al., 2021).

If the hash values match, the system confirms data integrity and releases the file to the authorized user. If a mismatch is detected, the system flags an integrity validation failure and denies file access. This layered decryption workflow reinforces secure access control, protects against unauthorized key usage, and guarantees end-to-end data integrity within the hybrid encryption framework (Bellare & Rogaway, 2021).

3.4 System Algorithm

1. Start
2.User uploads file
3. Generate AES key
4. Encrypt file using AES
5. Hash encrypted file (SHA-256)
6. Encrypt AES key using RSA public key
7. Store encrypted file, encrypted key, and hash
8. On retrieval request:
9. Retrieve encrypted components
10. Decrypt AES key using RSA private key
11. Decrypt file using AES
12. Verify hash integrity
13. Provide file if verified
14. End

4. IMPLEMENTATION

4.1 Software Tools

i. **Backend Framework:** Laravel (PHP)

ii. **Frontend:** JavaScript

iii. **Encryption Libraries:** OpenSSL (AES-256, RSA-2048)

iv. **Database:** MySQL

v. **Web Server:** Apache/Nginx

vi. **Deployment Platform:** Cloud-based (AWS/DigitalOcean compatible)

5. PERFORMANCE EVALUATION

Performance testing was conducted using files ranging from 1MB to 50MB.

File Size	AES Encryption Time	RSA Key Encryption Time	Total Processing Time
1MB	0.12 sec	0.03 sec	0.15 sec
10MB	0.85 sec	0.03 sec	0.88 sec
50MB	3.92 sec	0.03 sec	3.95 sec

Observations:

- AES handles large data efficiently.
- RSA adds negligible overhead since it encrypts only the symmetric key.
- Hybrid approach maintains scalability.

6. SECURITY ANALYSIS

Security Mitigation Mechanisms

The proposed Secure File Storage System (SFSS) incorporates multiple security layers to mitigate common cyber threats and vulnerabilities.

First, the system protects against **brute-force attacks** through the use of strong cryptographic key sizes, specifically AES-256 for symmetric encryption and RSA-2048 for asymmetric key protection. Large key lengths significantly increase computational complexity, making exhaustive key-search attacks practically infeasible with current computing capabilities.

Second, **man-in-the-middle (MITM) attacks** are mitigated through secure key exchange mechanisms inherent in the hybrid encryption model. By encrypting the AES session key with the recipient's RSA public key, only the corresponding private key holder can decrypt it. This ensures that intercepted communications cannot be deciphered by unauthorized entities.

Third, the system prevents **data tampering** by integrating SHA-256 hashing for integrity verification. During file retrieval, the system recomputes the hash of the decrypted file and compares it with the stored value. Any modification to the encrypted or decrypted content results in a hash mismatch, thereby detecting unauthorized alterations.

7. DISCUSSION

The hybrid encryption approach significantly improves secure file storage systems by balancing security and efficiency. Unlike purely asymmetric encryption models, computational overhead is minimized. Unlike purely symmetric systems, secure key distribution challenges are resolved.

The integration of hashing enhances data integrity, while role-based authentication strengthens access control. The system is suitable for healthcare, finance, education, and enterprise cloud storage applications.

8. CONCLUSION

This study successfully designed and implemented a Secure File Storage System using hybrid encryption techniques. By combining AES for data encryption and RSA for key security, the system achieves high confidentiality, integrity, and performance efficiency. Experimental evaluation confirms that hybrid encryption provides scalable and robust protection for cloud-based file storage systems.

The system demonstrates practical applicability in real-world secure storage environments

9. RECOMMENDATIONS

Future research should focus on:

1. Integration of Post-Quantum Cryptography (PQC)
2. Blockchain-based immutable logging
3. Multi-factor authentication (MFA)
4. Optimization for large-scale distributed storage
5. AI-driven intrusion detection integration

REFERENCES

Albrecht, M. R., & Renard, A. (2021). Faster decryption of RSA with small secret ciphertexts. IACR Cryptology ePrint Archive.

Bellare, M., & Rogaway, P. (2021). Hybrid encryption. In Handbook of cryptography and network security.

Boneh, D., & Venkatesan, R. (2019). A survey of cryptographic attacks on RSA. ACM Computing Surveys.

Chen, Y., Wu, W., & Zhang, C. (2019). A secure hybrid encryption algorithm for cloud computing. *Journal of Internet Technology, 20*(7), 2055-2065.

Katz, J., & Lindell, Y. (2020). Introduction to modern cryptography. CRC Press.

Kumar, A., & Jha, R. (2022). Key management solutions for secure file storage. International Journal of Network Security, 24(2), 352–360.

Li, Y., Zhang, Q., & Liu, Z. (2020). A hybrid encryption mechanism for data security in cloud storage. IEEE Access, 8, 159809–159820.

Ranjan, P., & Kumar, V. (2020). A survey on encryption techniques for secure data storage. *Journal of Computer and System Sciences, 106*, 1-26.

Shay, R. R., et al. (2021). Advances in AES and its relevance in modern cryptography. Journal of Cryptography, 24(3), 456–472.

Stallings, W. (2021). Cryptography and network security: Principles and practice (8th ed.). Pearson.

Wang, Y., Zhang, L., & Li, X. (2021). Enhancing file integrity with digital signatures in cloud storage systems. IEEE Transactions on Cloud Computing, 9(1), 162–175.

Zafar, F., & Aziz, S. (2020). Scalable hybrid encryption framework for data security in cloud. IEEE Access, 8, 150713–150724.