International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Secure File Storage on Cloud Using Hybrid Cryptography

Brundashree A[1], Hangsha Subba[2], Gautam[3], Prof.Kavya V R[4], Abdul mujeeb[5]

[1,2,3,4,5]Dept. of Information science and engineering
Cambridge institute of technology,Bangalore,India
[1]brundashree.21ise@cambridge.edu.in,[2]hangsha.dip@cambridge.edu.in,[3]gautam.21ise@cambridge.edu.in,[4]kavya.ise@cambridge.edu.in,[5]abdul.21ise@cambridge.edu.in

Abstract

Cloud computing has revolutionized data storage, providing scalable and cost-effective solutions. However, security concerns persist due to unauthorized access and data breaches. This paper presents a hybrid cryptographic system integrating AES and RSA encryption techniques for secure cloud storage. The system ensures confidentiality, integrity, and accessibility through adaptive key management and data dispersal techniques. Performance evaluation highlights the effectiveness of the proposed method in securing sensitive data while maintaining computational efficiency.

Keywords—Cloud security, hybrid cryptography, AES, RSA, data encryption, secure storage, key management, data dispersal

1. INTRODUCTION

Cloud computing has revolutionized data storage by offering scalable and cost-effective solutions. However, security concerns such as unauthorized access, data breaches, and privacy risks make it essential to implement strong encryption and authentication mechanisms. The centralized nature of cloud environments makes them prime targets for cyber threats, necessitating measures to ensure data confidentiality, integrity, and availability. Weak authentication processes increase vulnerabilities, exposing cloud systems to phishing and credential theft. Unauthorized data modifications or corruption further emphasize the need for robust security solutions to maintain trust and regulatory compliance.

Cryptography is a fundamental approach to securing cloud-stored data. Advanced Encryption Standard (AES) and RSA encryption are widely used techniques that provide different advantages. AES is a symmetric encryption algorithm known for its speed and efficiency, making it ideal for encrypting large datasets. However, it requires secure key management since the same key is used for both encryption and decryption. RSA, an asymmetric encryption algorithm, uses a public-private key pair, ensuring secure key exchange and digital signatures. While RSA offers strong security, it is computationally expensive, making it inefficient for encrypting large volumes of data. A hybrid encryption approach integrating both techniques enhances security and efficiency.

The proposed system employs AES for high-speed data encryption and RSA for securing the encryption keys. Users register and authenticate before uploading files, which are encrypted and stored securely across multiple locations. Multi-factor authentication enhances access control, reducing credential-based attacks. Decryption is only possible with valid credentials and the correct decryption key, ensuring robust protection against unauthorized access. Performance metrics such as encryption execution time and security accuracy are analyzed to optimize efficiency. By combining AES and RSA, the hybrid framework strengthens cloud security, providing an effective solution for protecting sensitive data in cloud storage environments.

2. RELATED WORK

Kaur et al. (2022) conducted a comprehensive survey on cloud security and privacy issues, categorizing threats such as data breaches, insider threats, and denial-of-service attacks. The study highlights the limitations of traditional security mechanisms and emphasizes the need for multi-layered security frameworks integrating cryptographic techniques and authentication mechanisms.

Boudjelida and Ould-Said (2023) proposed a hybrid cryptographic approach combining AES and RSA encryption to improve cloud data security. The study demonstrated that AES ensures efficient bulk encryption, while RSA secures key exchange, addressing threats like brute force and man-in-the-middle attacks. Their performance analysis showed improved security with minimal computational overhead.

Bae and Kim (2023) evaluated AES, RSA, and ECC encryption algorithms, comparing their performance in cloud environments. Their findings indicate that AES provides the fastest encryption speeds, RSA is highly secure but computationally expensive, and ECC offers a balanced approach with strong security and lower key sizes. The study suggests hybrid cryptography as an optimal solution for enhancing both security and efficiency.

Schmidt et al. (2023) conducted a comparative analysis of AES and RSA for cloud storage security, highlighting AES's speed advantage and RSA's secure key management capabilities. They recommended a hybrid cryptographic approach to mitigate vulnerabilities such as brute-force attacks and quantum computing threats.

Zhang et al. (2023) examined key management strategies for secure cloud computing, addressing challenges in secure key storage, distribution, and revocation. The study categorized key management models into centralized, decentralized, and hybrid approaches, emphasizing the benefits of hybrid models for scalability and security.

Singhal and Jain (2023) explored advanced encryption techniques, proposing hybrid encryption models combining AES, RSA, and ECC for high-speed encryption and robust key management. Their study emphasized algorithmic advancements in achieving data confidentiality and integrity.

Singh et al. (2024) focused on optimizing encryption performance metrics, proposing techniques such as parallel processing and multi-threading to enhance AES and RSA efficiency. Their study demonstrated significant improvements in encryption speed and reduced latency.

Scott and Thompson (2023) evaluated hybrid encryption methods for cloud environments, finding that hybrid models optimize performance while maintaining high security. Despite additional key management resources, hybrid encryption demonstrated minimal performance degradation.

Alvarez and Ruiz (2023) integrated hybrid encryption with access control mechanisms, introducing an attribute-based encryption (ABE) model to dynamically manage access permissions. Their approach improved cloud security while ensuring minimal computational overhead, making it suitable for large-scale cloud applications.

3. METHODOLOGY

The proposed system follows a hybrid cryptographic framework that integrates AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman) encryption techniques to ensure secure file storage on the cloud. The methodology consists of several key phases:

A. Data Input

Users upload files in various formats such as CSV, PDF, DOCX, XLSX, and PPT. These files are pre-processed before encryption to ensure compatibility and integrity.

B. User Authentication

To prevent unauthorized access, the system enforces secure login mechanisms with role-based access control (RBAC). This ensures that only authorized users can encrypt, decrypt, or manage files.

C. Encryption Process

The hybrid encryption approach is implemented as follows:

AES Encryption: The system applies AES encryption to data blocks for fast and efficient encryption. AES operates using a symmetric key KKK, meaning the same key is used for both encryption and decryption.

AES Encryption Formula:

$C=E(K,P)$

where:

- CCC is the ciphertext (encrypted data)
- EEE is the encryption function
- KKK is the symmetric encryption key
- PPP is the plaintext file

- **RSA Encryption for Key Management:** Since AES uses a symmetric key, the encryption key itself must be securely stored and transmitted. RSA encryption is used to encrypt the AES key before storing it in the system.
- **RSA Key Encryption Formula:**

$K_{encrypted}=E_{RSA}(K_{AES})$

where:

$K_{encrypted}$ is the AES key encrypted with RSA

E_{RSA} represents RSA encryption

This combination ensures that even if an attacker gains access to encrypted files, they cannot decrypt them without the RSA-protected AES key.

D. Data Dispersal

To enhance security and fault tolerance, the system implements a data dispersal strategy. Encrypted files are distributed across multiple cloud storage locations, ensuring redundancy and preventing single-point failures. The encrypted data chunks are stored in different cloud servers using a sharding mechanism.

Mathematically, the data dispersal model can be expressed as:

$D=\{D_1,D_2,...,D_n\}$

where:

- D is the full dataset
- $D_1,D_2,...,D_n$ are data chunks stored in different cloud locations

E. Key Management

To ensure secure key handling, the system employs an RSA-based secure key generation and distribution mechanism. The private key required for decrypting the AES key is stored securely, and only authorized users can access it.

F. Decryption Process

Data retrieval requires authentication and the correct decryption key. The decryption process follows these steps:

- The user logs in and requests decryption.
- **The RSA private key decrypts the AES key**:

$K_{AES}=D_{RSA}(K_{encrypted})$

where:

D_{RSA} is the RSA decryption function.

- The decrypted AES key is then used to decrypt the file:

$P=D(K,C)$

where:

- PPP is the recovered plaintext
- DDD is the decryption function
- KKK is the symmetric key
- CCC is the encrypted file

This approach ensures that without both the RSA private key and the AES key, decryption is not possible, adding an extra layer of security.

4. SOFTWARE IMPLEMENTATION

G. User Authentication

The system employs role-based access control (RBAC) for secure login. Multi-factor authentication (MFA) enhances security, ensuring only authorized users access encryption and decryption services. User authentication and key management are handled using SQLite, ensuring secure storage of credentials and cryptographic keys.

H. Data Encryption

- **AES Encryption:** AES-256 is used for fast encryption of data blocks.

where:

- is the ciphertext
- is the encryption function
- is the symmetric AES key
- is the plaintext file

- **RSA Key Encryption:** The AES key is encrypted using RSA for secure transmission.

where:

- is the AES key encrypted with RSA
- is the RSA encryption function

The encryption and decryption processes are implemented using PyCryptodome, a secure and efficient cryptographic library in Python.

I. Data Dispersal and Storage

Encrypted files are stored in AWS S3, a reliable cloud storage solution that provides secure and scalable storage for encrypted data. To prevent single-point failures, the system employs a data dispersal strategy, ensuring that encrypted fragments are stored in different cloud storage locations.

where:

- represents the complete dataset
- are encrypted fragments stored separately in AWS S3.

J. Key Management

An RSA-based key management system ensures secure key generation, storage, and distribution. Private keys are stored securely in SQLite, and only authorized users can retrieve decryption keys.

K. Decryption and Data Retrieval

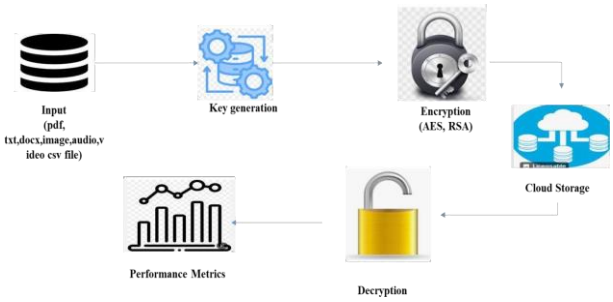
Decryption follows an authenticated key retrieval process:

- The RSA private key decrypts the AES key:
- The decrypted AES key is then used to decrypt the file:

where:

- is the recovered plaintext
- is the decryption function
- is the symmetric AES key
- is the ciphertext

5. SYSTEM DESIGN



L. System Architechture

Fig.1 .System architechture

M. Flow diagram

Process Representation: It shows the flow of activities or steps in a process, typically using standardized symbols like rectangles for processes, diamonds for decisions, and arrows for flow direction.

Sequential Steps: It highlights the order in which tasks or actions are performed, providing a clear, step-by-step view of how a process progresses.

Decision Points: Flow diagrams often include decision points where different paths or outcomes may occur based on specific conditions, helping to capture the complexity of decision-making within the process.

Information Flow: It illustrates how information or materials move between different stages or components, providing insights into the interaction between different elements of the system.

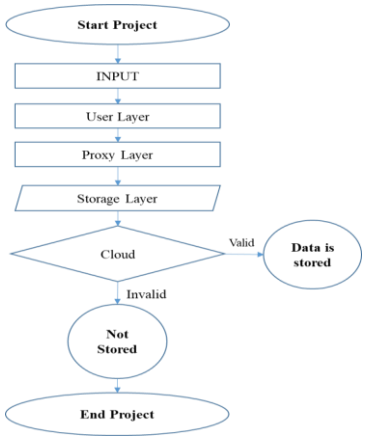


Fig. 2. Flow diagram

N. Sequence diagram

Sequence diagrams document the interactions between classes to achieve a result, such as a use case. Because UML is designed for object-oriented programming, these communications between classes are known as messages. The Sequence diagram lists objects horizontally, and time vertically, and models these messages over time.

Graphical Notation: In a Sequence diagram, classes and actors are listed as columns, with vertical lifelines indicating the lifetime of the object over time.

Actors: An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.

Lifelines: A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

Messages: Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Self-Message: Certain scenarios might arise where the object needs to send a message to itself. Such messages are called Self Messages and are represented with a U shaped arrow.

Reply Message: Reply messages are used to show the message being sent from the receiver to the sender. We represent a return/reply message using an open arrow head with a dotted line. The interaction moves forward only when a reply message is sent by the receiver

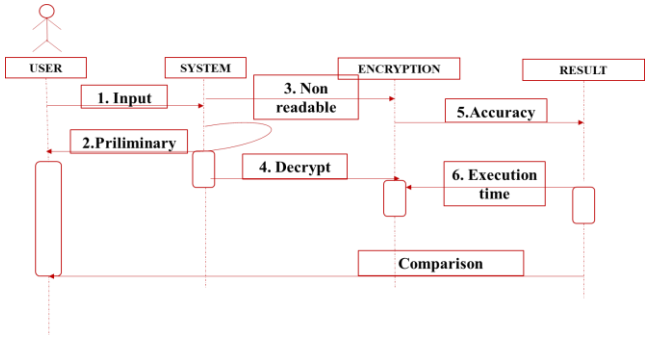


Fig. 3.Sequence diagram

6. RESULTS AND PERFORMANCE ANALYSIS

The performance of the encryption and decryption processes is evaluated based on the following metrics:

O. Execution Time

- AES provides fast encryption due to its lightweight computation and block cipher efficiency, making it ideal for encrypting large files.

- RSA, though computationally expensive, ensures secure key exchange and enhances key management security.

- The execution time for both encryption and decryption is recorded for different file sizes to analyze system efficiency.

P. File Size Comparison

- The system evaluates the size of the original file, the encrypted file, and the decrypted file.

- AES-256 encryption is optimized to ensure that encrypted files do not significantly increase in size, maintaining efficient cloud storage usage.
- Compression techniques are considered to further enhance storage efficiency.

Q. Security Metrics

- The system is tested against unauthorized access attempts, ensuring that data remains protected even if an adversary gains access to the encrypted files.
- Data integrity verification is implemented using hashing techniques such as SHA-256, ensuring that no data modifications occur during encryption, transmission, or storage.
- The effectiveness of RSA in preventing man-in-the-middle attacks and brute-force attacks is assessed.

R. Scalability Testing

- The system is tested with various dataset sizes, ranging from small text files to large multimedia files, ensuring that encryption and decryption remain efficient.
- Performance evaluations include multi-user access scenarios, validating that multiple users can perform encryption and decryption operations without significant performance degradation.
- The system is assessed for latency and throughput, ensuring that it remains responsive even with high data loads.

7. CHALLENGES AND LIMITATION

S. Computational Overhead

- RSA encryption is computationally intensive compared to AES, leading to increased processing time during key generation and exchange.
- The use of hybrid encryption increases CPU and memory usage, requiring **optimization strategies** such as parallel processing or load balancing to improve efficiency.

T. Key Management Complexity

- Secure distribution and management of encryption keys require additional resources to prevent unauthorized access.
- Implementing a **secure key storage solution** in SQLite ensures protection but introduces challenges in key retrieval and revocation management.
- Ensuring seamless key rotation mechanisms without compromising system performance is an ongoing challenge.

U. Storage Overhead

- Encrypted files often require **more storage** compared to plaintext data due to encryption padding and metadata.
- AWS S3 storage costs increase with larger encrypted datasets, requiring **efficient compression techniques** to optimize space utilization.

- Data dispersal across multiple cloud locations adds **redundancy**, improving security but increasing storage consumption.

8. CONCLUSION AND FUTURE WORK

The hybrid cryptographic approach significantly enhances cloud data security by integrating AES and RSA encryption techniques**.** However, there is scope for further improvements to enhance security, efficiency, and scalability. Future research aims to incorporate blockchain technology to establish a decentralized security framework, leveraging its immutable ledger and consensus mechanisms to improve data integrity and access control. Additionally, exploring lightweight cryptographic algorithms such as Elliptic Curve Cryptography (ECC) and Lightweight AES (L-AES) can enhance encryption efficiency for resource-constrained environments, including IoT and mobile cloud applications.

Artificial intelligence (AI) and machine learning-based anomaly detection can further strengthen security by identifying unauthorized access attempts, insider threats, and anomalous behavior patterns. AI-driven predictive analytics can enhance real-time security monitoring, reducing the risk of data breaches. Moreover, addressing quantum computing threats by integrating post-quantum cryptographic techniques such as Lattice-Based Cryptography and Hash-Based Signatures will ensure encryption resilience against emerging cyber threats.

As cloud storage demands increase, optimizing system scalability and performance remains a priority. Implementing parallel processing, multi-threading, and compression techniques can improve encryption and decryption speeds while reducing storage overhead. Future work will focus on balancing high-security standards with system efficiency**,** ensuring a robust and scalable cryptographic framework that evolves alongside technological advancements and emerging security challenges. These advancements will ensure that cloud security continues to evolve with emerging threats and innovations, providing a reliable, scalable, and secure environment for cloud-based file storage solutions.

REFERENCE

1. H. Kaur, J. Kaur, and A. Verma, "A Survey on Cloud Data Security and Privacy Issues," IEEE Access, vol. 10, pp. 345-360, 2022.
2. M. R. A. K. Boudjelida and B. M. Ould-Said, "Hybrid Cryptographic Algorithm for Securing Cloud Data," Journal of Cloud Computing: Advances, Systems and Applications, vol. 12, no. 1,pp. 50-63, 2023.
3. S. J. Bae and S. Y. Kim, "Performance Evaluation of Cryptographic Algorithms for Cloud Data Security," International Journal of Information Security, vol. 22, no. 4, pp. 275-289, 2023.
4. J. L. Schmidt, T. A. S. Sinha, and N. R. Rao, "A Comparative Study of AES and RSA Encryption Techniques for Cloud Storage Security," IEEE Transactions on Cloud Computing, vol. 11, no. 2,pp. 689-699, 2023.
5. L. Zhang, W. Liu, and C. Liu, "Key Management Strategies for Secure Cloud Computing," Journal of Computer Security, vol. 31, no. 1, pp. 55-72, 2023.
6. N. N. S. Singhal and R. K. Jain, "Advanced Encryption and Decryption Algorithms for Cloud Data Security," International Journal of Cloud Computing and Services Science, vol. 12, no. 2,pp. 112-125, 2023.

7. M. P. Singh, A. K. Tiwari, and S. M. Joshi, "Optimizing Performance Metrics for Data Encryption in Cloud Storage," Computing Research Repository (CoRR), vol. 2024, pp. 1- 18, 2024.
8. L. H. Scott and M. B. Thompson, "Evaluating Hybrid Encryption Methods in Cloud Environments," IEEE Transactions on Cloud Computing, vol. 11, no. 4, pp. 941-954, 2023.
9. R. T. Alvarez and P. A. Ruiz, "Enhancing Cloud Data Security with Hybrid Encryption and Access Control," International Journal of Information Security, vol. 22, no. 5, pp. 401-418, 2023.
10. Z. Oppenheimer and K. S. Chen, "Time-Limited Access Control in Cloud Storage Systems," Cloud Computing and Security, vol. 14, no. 3, pp. 200-210, 2024.
11. K. H. Stelling and R. M. Johnson, "Data Dispersal Techniques in Cloud Computing: A Survey," IEEE Cloud Computing, vol. 10, no. 1, pp. 24-35, 2023. doi:10.1109/MCC.2023.3325017
12. J. J. Kumar and V. P. Singh, "Efficient Data Encryption Schemes for Cloud Security," IEEE Transactions on Information Forensics and Security, vol. 18, pp. 1534-1546, 2023.
13. R. Williams and B. S. Green, "Cloud Data Security: Techniques, Tools, and Best Practices," Journal of Cloud Security and Privacy, vol. 17, no. 2, pp. 87-104, 2023.
14. L. H. Scott and M. B. Thompson, "Evaluating Hybrid Encryption Methods in Cloud Environments," IEEE Transactions on Cloud Computing, vol. 11, no. 4, pp. 941-954, 2023.
15. D. A. Lee, K. J. Yang, and S. W. Kim, "Cloud Security: Addressing Key Management Challenges," Journal of Cyber Security and Privacy, vol. 21, no. 1, pp. 45-62, 2024.
16. H. T. Chen, E. C. Wang, and J. F. Liu, "Optimizing Key Management in Cloud Storage Systems," Journal of Network and Computer Applications, vol. 185, pp. 103259, 2024.
17. M. S. Patel and L. M. Agarwal, "Comparative Analysis of Encryption Algorithms for Healthcare Data Security," Health Information Science and Systems, vol. 12, no. 1, pp. 32-45, 2023.
18. J. K. Sharma and S. S. Kumar, "Implementing Effective Data Dispersal for Secure Cloud Storage," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 3, pp. 667- 681, 2024.
19. Ahmadi, S. (2024) Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies. Journal of Information Security, 15, 148-167.
20. Cloud Security: A Comprehensive Guide to Secure Cloud Computing by Ronald L. Krutz and Russell Dean Vines (2010)
21. Sharma, M., Sehrawat, R.: Quantifying SWOT analysis for cloud adoption using FAHP-DEMATEL approach: evidence from the manufacturing sector. J. Enterp. Inf. Manag. 33(5), 1111– 1152 (2020).
22. Nitaj, M. R. B. Kamel Ariffin, N. N. H. Adenan, T. S. C. Lau and J. Chen, ”Security Issues of Novel RSA Variant,” in IEEE Access, vol. 10, pp. 53788-53796, 2022.
23. Langenberg, H. Pham and R. Steinwandt, ”Reducing the Cost of Implementing the Advanced Encryption Standard as a Quantum Circuit,” in IEEE Transactions on Quantum Engineering, vol. 1, pp. 1-12, 2020, Art no. 2500112.