SATBAYEV UNIVERSITY

Computing & Engineering

Volume 2 (2024), Issue 2, 1-7

https://doi.org/10.51301/ce.2024.i2.01

Secure File Storage on Cloud Using Hybrid Cryptography

E. Saginayev*, A. Niyazov, A. Razaque, Zh. Kalpeyeva, A. Urazgaliyeva

Satbayev University, Almaty, Kazakhstan

**Corresponding author: saginaev.eyn@mail.ru*

Abstract. This paper addresses the critical challenge of unau- thorized access to sensitive data stored on cloud platforms. As the volume of such data continues to grow, ensuring its confidentiality, integrity, and availability has become increasingly important. The research proposes a secure cloud storage solution that integrates advanced cryptographic techniques. Specifically, the study utilizes the Advanced Encryption Standard (AES- 256) for efficient data encryption and the RSA algorithm for secure key management, ensuring that only authorized users can access the data. To further enhance security, the Secure Hash Algorithm (SHA-256) is employed to verify data integrity by generating hash values before encryption and after decryption. The proposed solution demonstrates improved protection against unauthorized access and data tampering, providing a robust framework for cloud-based file storage. Through the integration of these methodologies, this research contributes to the ongoing efforts to secure sensitive data in cloud environments, highlighting the importance of adopting advanced cryptographic measures in the face of evolving cybersecurity threats.

Keywords: *cloud security, hybrid cryptography, AES-256, RSA Algorithm, SHA-256, data integrity, secure cloud storage.*

1. Introduction

The proliferation of cloud computing has revolutionized data storage and management, offering unprecedented scalability and flexibility for organizations worldwide. However, this shift towards cloud-based solutions has introduced significant security challenges, particularly concerning unauthorized access to sensitive data stored on cloud platforms. Studies have shown that cloud environments are vulnerable to cyberattacks and data breaches due to their multi-tenant architecture and shared resources [1]. Traditional encryption methods, such as DES and 3DES, are no longer sufficient to mitigate these risks, necessitating the adoption of advanced cryptographic approaches [2].

Symmetric encryption algorithms, like AES, are highly efficient for encrypting large datasets but present challenges in key management [3]. Conversely, asymmetric algorithms, such as RSA, address key distribution issues but are computationally intensive [4]. Hybrid cryptography, which combines these techniques, has emerged as a robust solution for secure data storage. For instance, combining AES for data encryption with RSA for secure key exchange offers both efficiency and security, making it a popular choice for cloud environments [5-6].

Moreover, cryptographic hashing algorithms, such as SHA-256, are widely used to ensure data integrity by detecting unauthorized modifications during transmission and storage [7]. By integrating these techniques, researchers aim to address the limitations of isolated methods, providing comprehensive security solutions. The integration of multi-factor authentication (MFA) further strengthens these frameworks by enhancing user authentication and reducing vulnerabilities associated with compromised credentials [8].

This research builds on the advancements in hybrid cryptographic systems to develop a secure file storage framework for cloud computing, emphasizing the importance of combining AES, RSA, and SHA-256 to ensure data confidentiality, integrity, and authentication.

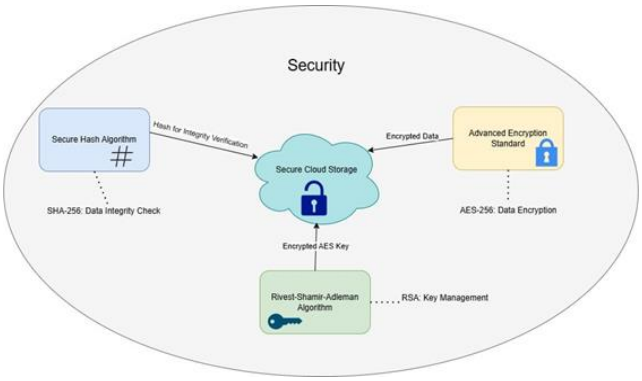

Figure 1. Hybrid encryption architecture for secure data storage in the cloud

1.1. Abbreviations

AES: Advanced Encryption Standard;
RSA: Rivest–Shamir–Adleman Algorithm;
SHA-256: Secure Hash Algorithm 256-bit;
DES: Data Encryption Standard;
3DES: Triple Data Encryption Standard;
MFA: Multi-Factor Authentication;
ECC: Elliptic Curve Cryptography.

2. Materials and methods

2.1. Research novelty and refined contribution

This research introduces a novel hybrid cryptographic framework that synergistically combines symmetric and asymmetric encryption techniques with cryptographic hash-

ing to secure cloud-based file storage systems comprehensively. Unlike previous studies that often focus on a single cryptographic method, our proposed solution integrates AES-256 for data encryption, RSA for secure key management, and SHA-256 for data integrity verification.

Refined Contributions:

Unified Security Model: By employing a combination of AES-256, RSA, and SHA-256, we provide confidentiality, key security, and data integrity all in one framework.

Efficiency-Driven Design: We demonstrate that the hybrid approach can encrypt large datasets efficiently (due to AES) while leveraging RSA only for key management, thus minimizing computational overhead.

Scalability and Adaptability: The model is designed to accommodate various file sizes and can be integrated into existing cloud architectures with minimal modifications. Integrity Assurance: Incorporating SHA-256 hashing ensures real-time detection of tampering, strengthening trust in cloud storage systems.

2.2. Proposed hypothesis and proof outline

Hypothesis Statement: «The proposed hybrid cryptographic framework (AES-256 + RSA + SHA-256) ensures a more secure and computationally efficient mechanism for cloud file storage compared to using purely symmetric or purely asymmetric cryptography, while maintaining robust data integrity».

A. Proof Sketch and Mathematical Formulation

We present a simplified proof approach, focusing on security and computational efficiency.

1) Security:

Let A denote the symmetric encryption algorithm (AES-256) and B denote the asymmetric encryption algorithm (RSA). Security of AES-256 relies on the hardness of recovering the key *K*(AES) from ciphertext *E(F)* without brute force attacks exceeding feasible computational limits. RSA security relies on the intractability of factoring large integers (at least 2048 bits).

Hence, an adversary must break both *A* and *B* to retrieve the plaintext file *F*. Probability of success is exponentially small if both algorithms are configured with standard security parameters.

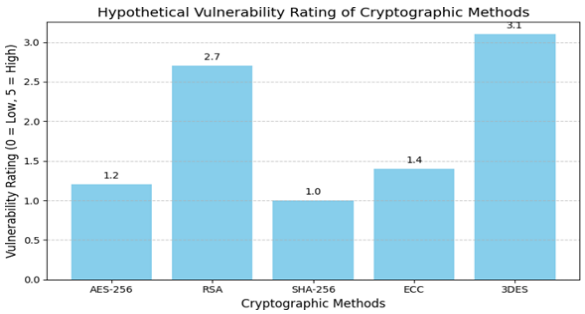

Figure 2. An illustrative chart comparing hypothetical vulnerability ratings for various cryptographic methods

As shown in Figure 2, each cryptographic method may exhibit a different level of vulnerability.

2) Computational Efficiency:

Define $T_A(n)$ as the time complexity of encrypting a file of size n bits with a symmetric key, and T_B as the time for encrypting a fixed-size AES key with an asymmetric algorithm.

Because *n* can be large (megabytes or gigabytes), symmetric encryption dominates the total encryption time, i.e.,

$$T_{\text{encrypt}} \approx T_A(n) + T_B(\text{keysize}), \quad (1)$$

where TB(keysize) is relatively small because it handles only a short key (e.g., 256 bits for AES).

Thus, the hybrid scheme is O(n) for large files, similar to pure AES, with negligible overhead from RSA key encryption.

3) Data Integrity:

Using SHA-256, define the hash function *H* such that

$$\text{hash} = H(E(F)) \quad (|\text{hash}| = 256 \text{ bits}) \quad (2)$$

The collision resistance of SHA-256 implies that finding two different ciphertexts *E(F)* and *E'(F)* that produce the same hash is computationally infeasible.

Therefore, any tampering or unauthorized modifications to *E(F)* will be detected during integrity checks.

Combining these arguments, we conclude that the proposed hybrid scheme is both secure (due to combined hardness of AES-256 and RSA) and computationally efficient (dominated by linear-time symmetric encryption).

2.3. Problem identification and significance

The critical issue addressed in this research is the vulnerability of cloud-stored data to unauthorized access and tampering due to inadequate encryption practices and weak key management. As organizations increasingly rely on cloud services, the potential impact of data breaches escalates, leading to financial losses, reputational damage, and legal consequences.

Key points include:

- Ensuring that confidential data remains secure from unauthorized entities.

- Preserving the trust of clients and stakeholders by demonstrating a commitment to robust data security practices. Adhering to legal and regulatory requirements regarding data protection and privacy.

- Preventing disruptions caused by security breaches that can affect organizational operations.

2.4. Problem solution

The proposed solution for secure cloud file storage leverages a hybrid cryptographic framework that effectively addresses the challenges of data security and performance. This model integrates symmetric and asymmetric cryptography, ensuring both rapid data processing and robust key management.

In the encryption phase, sensitive files are encrypted using a symmetric algorithm, specifically AES-256, known for its efficiency in handling large datasets. Following this, the sym- metric key utilized for AES encryption is securely encrypted with RSA. This two-tiered approach safeguards the key ex- change process, protecting it from potential interception by unauthorized parties.

To further enhance the security of stored data, the solution incorporates integrity checks through cryptographic hashing (SHA-256). This mechanism ensures that any unauthorized alterations to the data can be detected promptly, thus maintaining data integrity throughout its lifecycle. By combining these methodologies, the proposed solution establishes a comprehensive framework that meets the dual demands of speed and security in modern cloud environments.

2.5. Related works

The advent of cloud computing has revolutionized data storage and management, offering scalable and flexible solutions for individuals and organizations. However, this shift has also introduced significant security challenges, particu-

larly in safeguarding sensitive data from unauthorized access and breaches. To address these concerns, researchers have explored hybrid cryptographic approaches that combine symmetric and asymmetric encryption techniques to enhance data security in cloud environments. This section reviews several state-of-the-art methods that utilize hybrid cryptography for secure file storage in the cloud.

Satish B. Basapur et al. [1] proposed a hybrid cryptographic model integrating the Advanced Encryption Standard (AES) and the Rivest–Shamir–Adleman (RSA) algorithm to preserve sensitive data privacy. Their approach involves using AES- 256 for encrypting data due to its efficiency and speed, while RSA is employed to securely encrypt the AES symmetric key, enhancing key distribution security. To further strengthen key management, a multilayer perceptron (MLP) neural network is utilized for key generation and exchange. Sensitive attributes within the data are identified using the Depth First Search (DFS) technique, allowing for targeted encryption and masking of sensitive information. The model was evaluated using a credit card client's dataset, achieving an overall accuracy of 95.23.

Ritik Tyagi et al. [2] developed a secure file storage system using hybrid cryptography on the cloud by combining AES and RSA algorithms. In their system, data files are encrypted using AES to leverage its high performance in handling large datasets. The AES symmetric key is then encrypted using RSA, addressing the key distribution problem inherent in symmetric encryption methods. This dual-layer encryption ensures that even if unauthorized entities access the encrypted data, they cannot decrypt it without the corresponding RSA private key. The proposed architecture not only enhances data security during storage and transmission but also offers a scalable and efficient solution suitable for enterprises aiming to protect sensitive information in the cloud. Birendra Kumar Saraswat et al. [3] introduced a hybrid cryptographic approach that enhances cloud storage security by employing multiple encryption algorithms—AES, Data Encryption Standard (DES), and RC6. Their method involves splitting files into three parts (file slicing), each encrypted with a different algorithm. This strategy increases security by ensuring that even if one part is compromised, the others remain secure. User-specific keys are generated during registration and stored using steganography within the user's profile image, adding an additional layer of security for key management. Upon retrieval, the encrypted parts are decrypted using their respective algorithms and recombined to reconstruct the original file. This approach effectively mitigates risks associated with single-point encryption failures and enhances overall data security in the cloud.

Fortine Mata et al. [4] addressed cloud data security by proposing an enhanced encryption model that combines AES and Blowfish algorithms. Their model allows users to select the desired level of security, with different encryption algorithms applied based on this choice. By integrating AES, known for its strong security features, and Blowfish, recognized for its speed and efficiency, the model achieves a balance between security and performance. The hybrid approach increases the complexity for potential attackers due to the use of multiple encryption layers. The authors evaluated the model based on performance metrics such as throughput, encryption time, cipher text size, and delay. Results indicated that while the hybrid encryption increases processing time, it significantly enhances data confidentiality and integrity.

Jeganathan C et al. [5] proposed a secure file storage system in cloud computing utilizing a hybrid cryptography approach with AES and RSA algorithms. The system provides a user-friendly web portal interface for uploading and downloading files securely. When a user uploads a file, it is encrypted using AES for data confidentiality. The AES key is then encrypted using RSA before being stored in the cloud, ensuring that only authorized users with the corresponding RSA private key can decrypt the AES key and access the data. This method effectively secures the data during storage and transmission, and the integration of the web portal enhances usability without compromising security. Table I compares key approaches.

2.6. Proposed Hybrid Cryptographic Framework

This section provides a detailed overview of the proposed solution for achieving secure cloud-based file storage. The focus of this framework is to address critical challenges such as data confidentiality, integrity, and effective key management through the integration of advanced cryptographic techniques. Specifically, the framework leverages the strengths of AES- 256 for efficient data encryption, RSA for robust and secure key management, and SHA-256 for reliable integrity verification, ensuring a comprehensive security model.

A. Hybrid Cryptographic Framework Design

The proposed solution employs a hybrid cryptographic framework that seamlessly combines symmetric encryption, asymmetric encryption, and cryptographic hashing into a unified system. As conceptually illustrated in Figure 1 (referenced in the Introduction), this framework relies on three core cryptographic components, each contributing a distinct layer of security:

1) AES-256: This advanced symmetric encryption algorithm is particularly well-suited for encrypting large files. It offers an optimal balance between high-speed performance and strong resistance against cryptographic attacks, making it a reliable choice for securing cloud- stored data.

2) RSA: As an asymmetric encryption technique, RSA provides a secure mechanism for managing key exchanges. By encrypting the AES symmetric key, RSA ensures that the critical key material remains protected during transmission or storage, safeguarding it against unauthorized access.

3) SHA-256: Serving as a robust cryptographic hashing function, SHA-256 ensures the integrity of data by generating a unique hash value. This hash acts as a digital fingerprint, making it possible to detect any unauthorized modifications to the stored file, thereby preserving trust in the data's authenticity.

B. Algorithm for Secure Cloud Storage

To implement the proposed solution effectively, the framework employs a structured algorithm that integrates AES-256, RSA, and SHA-256 into a cohesive and efficient workflow. This algorithm is designed to address all critical aspects of secure cloud storage, from encryption and key management to integrity verification.

C. Algorithm for Secure Cloud Storage

The implementation of the proposed hybrid cryptographic framework adheres to a well-defined sequence of steps, ensuring seamless integration of AES, RSA, and SHA-256 processes. This workflow ensures a secure, efficient, and reliable process for encrypting, storing, and accessing sensitive data in cloud environments. By combining the computational efficiency of AES, the secure key management capabilities of RSA, and the tamper-detection strengths of SHA-

256, the proposed framework offers a robust solution to the challenges of secure cloud storage. The following algorithm provides a detailed breakdown of the workflow:

Algorithm: secure file storage process

1. Initialization: {F: Plaintext file; PK(RSA): RSA Public Key; SK(RSA): RSA Private Key; K(AES): AES Symmetric Key; E(F): Encrypted file; E(K(AES)): Encrypted AES Key; H(SHA): Hash of Encrypted File; C: Cloud Storage}
2. Input: {F, PK(RSA), SK(RSA)}
3. Output: {E(F), E(K(AES)), H(SHA)}
4. Set K(AES): Generate AES symmetric key for data en- cryption
5. Encrypt File F: E(F) = AES Encrypt(F, K(AES))
6. Encrypt Key K(AES): E(K(AES)) = RSA Encrypt(K(AES), PK(RSA))
7. Generate Hash H(SHA): H(SHA) = SHA256(E(F))
8. Store in Cloud: Transfer E(F), E(K(AES)), and H(SHA) to C

Decryption Process:

9. Retrieve from Cloud: Retrieve E(F), E(K(AES)), and H(SHA) from C
10. Decrypt Key K(AES): K(AES) = RSA Decrypt(E(K(AES)), SK(RSA))
11. Verify Integrity: Recalculate Hash H(SHA): H(SHA) = SHA256(E(F)). Compare H(SHA) with H(SHA)
12. Decrypt the File: Check If H(SHA) = H(SHA), F = AES Decrypt(E(F), K(AES))

Algorithm 1 starts with the initialization phase, which defines the core elements required for secure file encryption and storage. Step 1: Initialization. The original file (F) repre- sents the data to be protected. A symmetric key (K(AES)) is generated to encrypt the file using AES, chosen for its efficiency and high security. For key management, an RSA public key (PK(RSA)) and private key (SK(RSA)) pair are employed to ensure secure exchange of the symmetric key. Additional elements include the encrypted file (E(F)), the encrypted symmetric key (E(K(AES))), and a hash value (H(SHA)) generated using SHA-256 for integrity verification. The cloud storage (C) serves as the repository for storing encrypted artifacts. These components guarantee that data re- mains secure, tamper-proof, and accessible only to authorized users. Step 2: Input. Inputs include the plaintext file (F) to be encrypted, the RSA public key (PK(RSA)) for encrypting the symmetric key, and the RSA private key (SK(RSA)) for decrypting the symmetric key. Step 3: Generate AES Key.

Table 1. Comparison of modern methods based on related works

Methods/Approaches	Solutions	Advantages/Features	Limitations
Basapur et al. [1] Hybrid AES and RSA with Neural Network	AES-256 for data encryption, RSA for key encryption, MLP neural network for key generation and exchange	High accuracy (95.23%), Efficient key management, Targeted masking of sensitive data, Enhanced security through neural networks	Increased complexity due to neural networks, Potential computational overhead
Tyagi et al. [2] Hybrid AES and RSA	AES for encrypting data files, RSA for secure key exchange, Dual-layer encryption model	Enhanced data security during storage and transmission, Addresses key management challenges, Scalable and efficient for large datasets	Computational intensity of RSA may impact performance, Requires secure RSA key management
Saraswat et al. [3] Hybrid AES, DES, RC6 with Steganography	File slicing and encryption with AES, DES, RC6, User-specific keys stored via steganography	Increased security through multi-algorithm encryption, Secure key storage using steganography, Distribution across different cloud nodes	Increased complexity and processing time, Dependence on steganography may introduce vulnerabilities
Mata et al. [4] Hybrid AES and Blowfish	Data encryption using a combination of AES and Blowfish, User-selected security levels	Enhanced confidentiality and integrity, Combines strengths of both AES and Blowfish algorithms	Additional processing time due to hybrid encryption, may require more computational resources
Jeganathan et al. [5] Hybrid AES and RSA	AES for data encryption, RSA for encrypting AES key, Web portal interface for secure file management	User-friendly interface, Strong data security and confidentiality, Efficient encryption and decryption processes	Potential performance issues with RSA encryption, Security depends on the robustness of the web portal

A symmetric key (K(AES)) is generated to encrypt the file. Step 4: Encrypt File. The plaintext file (F) is encrypted using the AES symmetric key: E(F) = AES Encrypt(F, K(AES)). Step 5: Encrypt AES Key. The symmetric key (K(AES)) is encrypted with the RSA public key (PK(RSA)): E(K(AES))= RSA Encrypt(K(AES), PK(RSA)). Step 6: Generate Hash. The integrity hash (H(SHA)) is computed from the encrypted file: H(SHA) = SHA256(E(F)). Step 7: Store in Cloud. The encrypted file (E(F)), the encrypted symmetric key (E(K(AES))), and the hash value (H(SHA)) are stored in the cloud. Step 8: Retrieve from Cloud. The encrypted file (E(F)), the encrypted symmetric key (E(K(AES))), and the hash (H(SHA)) are retrieved from the cloud. Step 9: Decrypt AES Key. The RSA private key (SK(RSA)) is used to decrypt the AES symmetric key: K(AES) = RSA Decrypt(E(K(AES)), SK(RSA)). Step 10: Verify Integrity. The hash of the encrypted file is recalculated: H'(SHA) = SHA256(E(F)). The recalculated hash (H'(SHA)) is compared with the original hash (H(SHA)). If they match, the data integrity is verified. Step 11: Decrypt File. If the hash values match, the encrypted file (E(F)) is decrypted using the AES symmetric key: F = AES Decrypt(E(F), K(AES)).

The structured workflow begins with the input of the plaintext file (F) and the RSA key pair (PK(RSA), SK(RSA)). The output includes the encrypted file (E(F)), ensuring data confidentiality; the encrypted symmetric key (E(K(AES))), enabling secure key management; and the hash value (H(SHA)), verifying the integrity of the data. This step-by-step approach ensures robust security by combining encryption, secure key exchange, and data integrity verification. The process is scalable and protects sensitive data throughout its lifecycle in the cloud.

2.7. Key equations

Key Equations with Numbers:

$$E(F) = \text{AES\_Encrypt}\ F, K(\text{AES}) \quad (3)$$

Equation (3) represents the encryption of the original file F using the AES symmetric key K(AES). The output $E(F)$ is the ciphertext version of F.

$$E\ K(\text{AES}) = \text{RSA\_Encrypt}\ K(\text{AES}), PK(\text{RSA}) \quad (4)$$

Equation (4) describes how the AES symmetric key K(AES) is encrypted with the RSA public key PK(RSA). The result is an encrypted key E(K(AES)), which can only be decrypted by the corresponding RSA private key.

$$H(\text{SHA}) = \text{SHA256}\ E(F) \quad (5)$$

Equation (5) shows the process of creating a hash value H(SHA) for the encrypted file *E(F).* Using SHA-256 ensures that any unauthorized modifications to *E(F)* can be detected by comparing this hash value before and after storage or transmission.

$$K(\text{AES}) = \text{RSA\_Decrypt}\ E(K(\text{AES})), SK(\text{RSA}) \quad (6)$$

Equation (6) represents the decryption of the AES symmet- ric key K(AES) using the RSA private key SK(RSA). This step is necessary so that the decrypted AES key can later be used to decrypt the file.

Finally, Equation (7) shows the decryption of the ciphertext E(F) back into the original file F. This decryption uses the AES key K(AES) that was just recovered by the RSA decryption in the previous step.

Equations (3)–(7) together illustrate the entire process of encrypting and decrypting a file using a hybrid cryptographic approach. Specifically, AES is used for the file encryption/decryption steps, while RSA manages the secure exchange of the AES key. SHA-256 provides a reliable method for verifying the integrity of the encrypted data, thereby protecting against unauthorized modifications.

3. Results and discussion

3.1. Implementation and testing results

A. Experimental Setup

To validate the efficiency and robustness of the proposed Hybrid Cryptographic System, we conducted experiments in a controlled environment:

Programming Language: Java (JDK 17) Development Environment: IntelliJ IDEA 2023 Operating System: Windows 11 (64-bit) Hardware Configuration:

– CPU: Intel Core i7 (12th Gen), 3.6 GHz

– RAM: 16 GB DDR4

– Storage: 512 GB SSD

Libraries Used: Java Cryptography Architecture (JCA), Bouncy Castle

B. File Encryption and Decryption Times

We tested performance using files of varying sizes (1 MB, 5 MB, 10 MB, 50 MB). Figure 3 and Table II show the results.

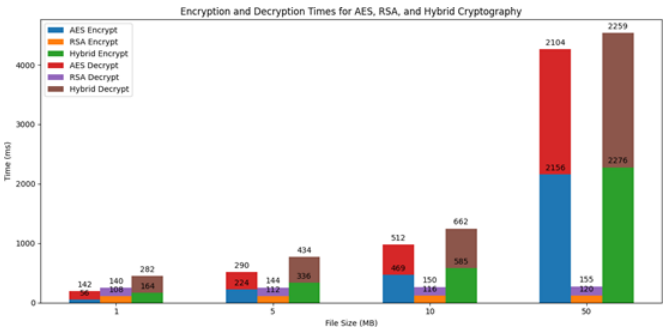

Figure 3. Encryption and Decryption times for AES, RSA, and Hybrid Cryptography across varying file sizes

Table 2. Encryption and Decryption Times for Various File Sizes

File Size (MB)	AES Enc (ms)	RSA Key Enc (ms)	Total Enc (ms)	Dec (ms)
1	56	108	164	142
5	224	112	336	290
10	469	116	585	512
50	2156	120	2276	2104

AES encryption demonstrates near-linear scalability with file size.

RSA key encryption adds moderate overhead but does not significantly impact total encryption time for large files.

C. Data Integrity Verification

To test the reliability of SHA-256 for data integrity, intentional modifications were introduced in the encrypted files. Table 3 shows the integrity check results.

Table 3. Data Integrity Verification Results

File Size (MB)	Tampered File (Y/N)	Integrity Status
1	Yes	Failure
5	No	Success
10	Yes	Failure
50	No	Success

The results demonstrate that the hybrid cryptographic framework, integrating AES-256 for data encryption, RSA for key management, and SHA-256 for integrity checks, is both secure and efficient.

Comparative Analysis:

Efficiency and Speed: Compared to purely asymmetric methods, our approach significantly reduces encryption time for large files.

Secure Key Management: RSA resolves the key distribution challenge inherent in symmetric encryption.

Data Integrity: SHA-256 reliably identifies tampering or corruption.

Shortcomings:

RSA still adds some overhead for key encryption, but the trade-off is justified by stronger key management.

For extremely large files (hundreds of MB to GBs), optimizing AES parameters or exploring ECC for key management could further improve performance.

4. Conclusions

In this study, we proposed and validated a novel hybrid cryptographic framework for secure file storage in cloud environments. The framework effectively integrates AES-256 for data encryption, RSA for secure key management, and SHA-256 for integrity verification.

A. Key Findings

Performance Efficiency: AES-256 provides fast and scalable encryption for large datasets.

Secure Key Management: RSA ensures secure exchange of the AES key, addressing a critical vulnerability in symmetric-only schemes.

Data Integrity: SHA-256 hashing robustly detects tampering, safeguarding stored data against unauthorized alterations.

B. Comparison with Existing Work

While existing hybrid solutions may focus on either symmetric or asymmetric techniques, the proposed system offers a comprehensive approach—balancing encryption speed, secure key distribution, and data integrity checks in a single package.

C. Future Scope

Future research may explore:

Incorporating ECC instead of RSA to reduce key- management overhead and energy consumption.

Integrating advanced authentication mechanisms (e.g., bio- metric or MFA) for heightened user-level security.

Testing in distributed or multi-cloud environments to evaluate scalability and fault tolerance.

References

[1] Basapur, S.B., Shylaja, B.S. & Venkatesh. (2021). A Hybrid Cryptographic Model Using AES and RSA for Sensitive Data Privacy Preserving. *Special Issue on Current Trends in Management and Information Technology, (18).* https://doi.org/10.14704/WEB/V18SI05/WEB18219

[2] Tyagi, R. & Verma, R. (2024). Secure File Storage Using Hybrid Cryptography on Cloud. *International Journal of Research Publication and Reviews, 5(6).*

[3] Saraswat, B.K. (2023). Secure File Storage Using Hybrid Cryptography on Cloud. *International Journal of Research Publication and Reviews, 4(5), 5050–5053*

[4] Mata, F., Kimwele, M. & Okeyo, G. (2017). Enhanced Secure Data Storage in Cloud Computing Using Hybrid Cryptographic Techniques (AES and Blowfish). International Journal of Science and Research (IJSR), 6(3). https://doi.org/10.21275/ART20171804

[5] Verma, V., Kumar, P., Verma, R.K. & Priya, S. (2021). A Novel Approach for Security in Cloud Data Storage Using AES-DES-RSA Hybrid Cryptography. *Emerging Trends in Industry 4.0 (ETI 4.0).* https://doi.org/10.1109/ETI4.051663.2021.9619274

[6] Jeganathan, C. (2024). Secure File Storage Using AES & RSA Algorithm in Cloud Computing. *Journal of Emerging Technologies and Innovative Research (JETIR)*

[7] Sharma, V. (2021). Secure File Storage on Cloud using Hybrid Cryptography. *5th International Conference on Information Systems and Computer Networks (ISCON).* https://doi.org/10.1109/IS- CON52037.2021.9702323

[8] Chisoni, G. & Selvam, G.G. (2023). The Design and Implementation of a Secure File Storage on the Cloud using Hybrid Cryptography. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 3(1). https://doi.org/10.48175/IJARSCT-9067

[9] Susmitha, C. (2023). Hybrid Cryptography for Secure File Storage. *7th International Conference on Computing Methodologies and Communication (ICCMC-2023).* https://doi.org/10.1109/IC- CMC56507.2023.10084073

[10] Ghadi, A.S. (2020). Secure File Storage Using Hybrid Cryptography. *International Journal of Innovative Science and Research Technology, 5(12)*

[11] Lai, J.-F. & Heng, S.-H. (2022). Secure File Storage On Cloud Using Hybrid Cryptography. Journal of Informatics and Web Engineering, 1(2). https://doi.org/10.33093/jiwe.2022.1.2.1

[12] Betrand, C.U., Onukwugha, C.G., Benson-Emenike, M.E., Ofoegbu, C.I. & Awaji, N.M. (2024). File Storage Security in Cloud Computing Using Hybrid Encryption. *Internet of Things and Cloud Computing, 12(1), 1–9.* https://doi.org/10.11648/j.iotcc.20241201.11

[13] Fatima, S., Rehman, T., Fatima, M., Khan, S. & Ali, M.A. (2022). Comparative Analysis of AES and RSA Algorithms for Data Security in Cloud Computing. *Eng. Proc., 20(14).* https://doi.org/10.3390/eng- proc2022020014

[14] Ramdhani, M.D. & Fuad, A.A. (2020). A Study on Hybrid Cryptog- raphy Approach (AES-RSA) in Securing Cloud Environment. *Proceedings of the 3rd International Conference on Informatics and Computational Sciences (ICICoS), IEEE.* https://doi.org/10.1109/ICI-CoS51170.2020.9299003

Гибридті криптографияны қолдану арқылы бұлттағы файлдарды қауіпсіз сақтау жүйесі

Э. Сагинаев*, А. Ниязов, А. Разак, Ж. Кальпеева, А. Уразгалиева

Satbayev University, Алматы, Қазақстан

**Корреспонденция үшін автор: saginaev.eyn@mail.ru*

Аңдатпа. Бұл мақала бұлттық платформаларда сақталатын құпия деректерге рұқсатсыз қол жеткізу мәселесін қарастырады. Мұндай деректер көлемінің артуына байланысты олардың құпиялылығы, тұтастығы және қолжетімділігі күн санап маңызды бола түсуде. Зерттеу жұмысы жетілдірілген криптографиялық әдістерді біріктіре отырып, қауіпсіз бұлттық сақтау жүйесін ұсынады. Атап айтқанда, деректерді тиімді шифрлау үшін AES-256 алгоритмі қолданылады, ал рұқсат етілген пайдаланушылар ғана деректерге қол жеткізе алуы үшін RSA алгоритмі арқылы кілттерді басқару жүзеге асырылады. Қауіпсіздікті одан әрі күшейту мақсатында деректердің бұрмаланбауын тексеру үшін шифрлау алдында және шифрдан шығару кейін Secure Hash Algorithm (SHA-256) алгоритмі арқылы хэш мәндері жасалады. Ұсынылған шешім рұқсатсыз қол жеткізу мен деректердің өзгертілуіне қарсы сенімді қорғанысты қамтамасыз етіп, бұлттық файлдық сақтау жүйесі үшін берік негіз қалыптастырады. Осы әдістерді үйлестіру арқылы зерттеу бұлттық ортадағы құпия деректерді қорғау жөніндегі жұмыстарға үлес қосып, киберқауіпсіздік қатерлерінің үнемі өзгеріп отыратын жағдайында заманауи криптографиялық шараларды қолданудың маңыздылығын айқындайды.

Негізгі сөздер: *бұлттық қауіпсіздік, гибридті криптография, AES-256, RSA алгоритмі, SHA-256, деректер тұтастығы, қауіпсіз бұлттық сақтау.*

Безопасное хранение файлов в облаке с использованием гибридной криптографии

Э. Сагинаев*, А. Ниязов, А. Разак, Ж. Кальпеева, А. Уразгалиева

Satbayev University, Алматы, Казахстан

**Автор для корреспонденции: saginaev.eyn@mail.ru*

Аннотация. В данной статье рассматривается актуальная проблема несанкционированного доступа к конфиденциальным данным, хранящимся в облачных платформах. По мере увеличения объемов таких данных обеспечение их конфиденциальности, целостности и доступности становится все более важной задачей. В исследовании предлагается безопасное решение для облачного хранения файлов, основанное на передовых криптографических методах. В частности, для эффективного шифрования данных используется алгоритм AES-256, а для надежного управления ключами применяется алгоритм RSA, что гарантирует доступ к данным только авторизованным пользователям. Дополнительно для проверки целостности данных используется алгоритм хеширования SHA-256, который генерирует контрольные суммы до и после шифрования. Предложенное решение обеспечивает повышенную защиту от несанкционированного доступа и изменений данных, создавая надежную основу для облачного хранения информации. Интеграция этих методов способствует укреплению защиты конфиденциальных данных в облачных средах, подчеркивая важность применения современных криптографических технологий в условиях растущих киберугроз.

Ключевые слова: *облачная безопасность, гибридная криптография, AES-256, алгоритм RSA, SHA-256, целостность данных, безопасное облачное хранение.*

Received: 09 February 2024
Accepted: 15 June 2024
Available online: 30 June 2024