PENINSULA PUBLISHING LLC

SHIFRA
Vol. **(2026)**, 2026, **pp**. 97–106
ISSN: 3078-3186

Research Article

An Intelligent Cloud Computing Security Approach Using a Dual Hybrid Cipher System

Ayad Osama Jalal[1, *]

[1] *Faculty of Administration and Economics, Al-Iraqia University, Baghdad, Iraq*

ARTICLEINFO

Article History
Received 2 Jan 2026
Revised: 17 Feb 2026
Accepted 18 Mar 2026
Published 2 Apr 2026

Keywords
Dual Cipher System,
loud Computing Security,
Hybrid Encryption,
AES-RSA,
Cryptographic Framework,
Data Confidentiality.

ABSTRACT

Although the development of cloud computing has led to a paradigm shift in the data storage, it has also subjected the sensitive information to massive security and performance threats; that is, the data integrity and processing overhead. There is a basic trade-off in classical cryptography: the symmetric-key protocols are really fast, whereas they do not provide secure exchange of keys, but the asymmetric-key protocols are very secure and computationally infeasible when dealing with large amounts of data. Conventional cryptographic techniques commonly have a fundamental dilemma: symmetric-key protocols can offer fast execution but do not include any secure key-exchange protocols, whereas asymmetric-key systems can offer high protection but cannot be used with large volumes of data due to their slow computation speed. To alleviate these problems, this study presents a streamlined dual-cipher architecture that is aimed at boosting the efficiency of the cloud through the integration of the AES-256 and RSA-2048. The intelligence of the system lies in a dynamic workload distribution mechanism such that the high-resource encapsulation of the RSA operations is limited to the encryption of keys, while the bulk encryption is left to the high-speed AES engine. The empirical assessment of different file sizes (1 MB to 1 GB) shows that a sustainable throughput of 56-72 MB/s is achieved at 83-86 percent of the functional speed of standalone AES. As a result, the suggested model provides a viable and scalable design for modern cloud infrastructures with a very low overhead of just 20-21 percent or so relative to single deployments of RSA.

1. INTRODUCTION

With developments in different spheres of life and evolution of service, commercial, healthcare institutions, there has been a demand to keep abreast with this. This becomes especially significant in terms of updating information transfer processes and safeguarding them against hacking and other dangers through increasing their overall reliability [1]. Cloud computing was created to respond to these requirements and deal with the advancements in different industries [2]. The interest of this field on a global level has grown and companies and institutions are becoming more dependent on it particularly when it comes to safe storage and transfer of data, ensuring data confidentiality against unauthorized access [3]. Nonetheless, there have been a few hurdles and problems regarding cloud computing security, because the data of such companies and institutions is stored on servers that are not on their physical district. The high level of harm inflicted by data breaches amounted to almost $5 million USD in 2024, necessitating the development of efficient countermeasures and mitigation strategies [4]. To solve this issue, a robust cryptographic framework utilizing a number of encryption algorithms, both symmetric (e.g., AES, DES) and asymmetric (e.g., RSA, ECC) have been developed, associated with their own pros and cons [5]. The main feature of symmetric encryption is the high speed of encryption, especially when it comes to large files, but in the cloud environment it works very slowly because of the difficulties with key distribution between different users [6]. Conversely, asymmetric encryptions are better in handling key administration and allocation, although they are slower with bigger databases and they cost more to compute [7]. Other issues, including the insider threats by cloud service providers and the issues of latency, can be also mitigated by using strong encryption techniques that are properly combined as hybrid encryption schemes [8].

To attain the core aspects of cybersecurity, confidentiality, integrity, availability, authentication and access control are all requirements. As such, it is necessary to have certain security priorities whose most crucial ones are identity and access management, data encryption and security, infrastructure security, incident response and compliance [9]. This is what Cloud Computing Security Alliance refers to [10]. Cloud computing involves distributed responsibility where the cloud

*Corresponding author email: ayad.o.jalal@aliraqia.edu.iq
DOI: https://doi.org/10.70470/SHIFRA/2026/005

service providers are responsible of ensuring the security of the basic infrastructure and the clients are responsible of ensuring all the measures necessary to ensure security of their data and applications [11]. Encryption has been regarded as one of the primaries supports under which cloud computing systems have been founded to ensure that the data stored in them is not compromised or accessed by unauthorized individuals. This data encryption secures the confidentiality, integrity and availability of the data thus providing a secure collection, storage and transmission of data in cloud computing environments. Identity and access management methods, data encryption, and protection of infrastructures, incident response and adherence to standards are normally prioritized in proper cloud security [12].

Modern cryptographic techniques can be broadly classified into symmetric-key and asymmetric-key encryption algorithms [13]. A shared secret key is utilized for both encryption and decryption purposes. In practice, this key serves as a means of secure communication between two or more communicating parties. Several well-known symmetric encryption algorithms, such as AES and chacha20 under this category [14]. AES is the most used symmetric encryption algorithm for cloud data protection. This type of encryption uses bit keys sizes of 128, 192, or 256 bits. This encryption method is characterized by its ability to handle massive amounts of data with minimal Data-processing overhead. Therefore, it is used in modern applications that require high-speed data encryption, which can reach up to 10 gigabytes per second (using AES-NI instructions) [15]. ChaCha20 This type of encryption is a modern one that has been modified to provide better resistance and diffusion. It was improved by Daniel J. Bernstein based on variant Salsa20 [16] and works perfectly with mobile phone software. Because of its simplicity and speed, it is well-designed to work seamlessly with modern devices such as mobile phones, the Internet of Things, and cloud environments. It is used to terminate timing vulnerabilities that are widespread in old ciphers. Besides that, it is used as an alternative to AES and has been integrated into most TLS and SSH protocols. Overall, it is considered as most trusted stream cipher [17].

This encryption method utilizes a dual set of keys - one being public and the other private with the goal of enhancing security. These keys are mathematically interconnected, yet deciphering one from the other is a near-impossible feat. Participants maintain two keys: a publicly accessible key, visible to anyone on the network and thus not necessitating added security measures, and a confidential private key, known solely to its owner and kept undisclosed. Algorithms based on this method are used, for example, Diffie-Hellman exchange of keys, Rivest-Shamir-Adleman (RSA) and elliptic curve methods [18]. RSA With a computational complexity of large prime numbers, it has been applied to provide a secure digital signature and key exchange, due to the difficulty of factorizing large prime numbers. It uses 2048 bits as minimum keys as required (The typical length of the key range is between 3072-4096 bits) to provide higher security. The downside of this method is the substantial processing cost; this makes it completely impractical for encrypting large data. for that, it is usually used to encrypt only symmetric keys [19]. Elliptic Curve Cryptography (ECC) This type of encryption provides security comparable to RSA encryption using smaller keys, resulting in more efficient performance. ECDH is an application of ECC technology, offering secure key exchange. This method is used in cloud environments when a group of individuals needs to exchange a shared key without direct interaction [20].

There are several studies that have suggested the incorporation of hybrid encryption schemes with the combination of both symmetric and asymmetric encryption. Previous studies in this field involve: Bharti et al. [21] paid attention to the comparison of symmetric and asymmetric algorithms. Their efforts were to interpret the encryption models and promote more developed models to suit the contemporary needs of the life in the cloud computing environment to safeguard the data against numerous threats. Kumar and Kumar [22] focused on finding an algorithm for cloud computing platforms that improves the security and encryption of their data. They developed a hybrid algorithm modified with AES-ECC technology, and through this algorithm, they were able to reduce the time required for encryption and decryption, and they found an increase in the security level compared to regular algorithms. Shivaramakrishna and Nagaratna [23] used a hybrid AES-OTP and RSA encryption system designed to provide key management capabilities by specifying who can access the data, the duration of this access, and the time of access termination. These elements are so important in cloud environments because it is regarded as dangerous for data to remain exposed. Zhao et al. [24] in their research was demonstrated that "hybrid encryption", despite its effectiveness, it still shows difficulty in simultaneously achieving all the desired benefits, such as high efficiency, exceptional security, with fast data transfer. Table I provides a summary of the representative hybrid encryption strategies and their principal features, providing the contextual comparison and the offered system.

TABLE I. SUMMARY OF RELATED HYBRID CRYPTOGRAPHIC APPROACHES IN CLOUD COMPUTING

Ref#	Year	Algorithms Used	Application Domain	Key Results
[21]	2024	Symmetric & Asymmetric (Survey)	Cloud Security	Highlighted need for advanced hybrid models
[22]	2023	AES–ECC	Cloud Storage	Reduced encryption time and improved security
[23]	2023	AES–OTP + RSA	Cloud Data Access Control	Time-limited access and secure key management
[24]	2025	Hybrid Encryption & Compression	Secure Data Transmission	Identified trade-offs between efficiency and security
This Work	2025	AES–RSA	Intelligent Cloud Security	Balanced security and performance with ≤21% overhead

Even though there has been a progress in hybrid cryptographic schemes, much of the literature in the past has concentrated on either hypothetic security or on described complex access control systems that do not allow practical scaling and reproducibility in clouds. In addition, many of the existing models do not provide performance metrics that can be potentially forecasted when used in large data sets. To address the above gaps, this paper will suggest a better dual hybrid cipher architecture on a cloud environment that pays close attention to combining AES-256 (to achieve high-speed bulk encryption) and RSA-2048 (to encapsulate keys with high strength). The primary originality of the work is its scheme of resources allocation. Unlike the traditional static methodologies, the proposed system is a dynamic system: the computational workload is offloaded to an important management module, which restricts intensive public-key operations, and to a high-performance symmetric engine, which offloads large volumes of data. With this design, the overhead of the CPU and response time is minimized and this makes working with cloud structure very affordable. The key results of the research may be stated as follows:

1. Dual-Architecture Design: Designing a hybrid system that is uniquely designed to take advantage of cloud resources, in the form of greater security and performance.
2. Comprehensive Performance Benchmarking: Confirmation of the system by performing massive testing of the encryption response time, throughput, and CPU usage with a wide range of file sizes (1 MB to 1GB).
3. Optimized Key Distribution: Verifying the secure transmission of keys at low performance cost, achieving 83-86 percent of the native AES performance with overheads that are manageable at 20 -21 percent.
4. Practicality and Trade-off Analysis: A detailed evaluation of the trade-off between the cryptographic strength and the computational efficiency and the viability of the system in the actual application of cloud deployment to the real world.

In order to clearly build a contrast between this work and the usual implementations and dissimilar to the old method of having static workload distribution by making sure that the AES and RSA are not used as a dynamic workload, the present paper is implementing a dynamic workload distribution mechanism. The main technical advances in this respect include: a smart decision logic to limit the computational costly RSA computation only to key encapsulation, and a dual-cipher design to be able to achieve an optimal AES-256 throughput scale on large data sets; and a dual-cipher architecture to be able to trade-off between the hardness and the cost of security in cloud storage. We organized the parts of this work in the following way, section two elaborates and describes the proposed methodology, and the third section offers an in-depth performance analysis. Section four is the last part and it represents the conclusion of the paper and specifies the direction of research in the future.

2. RESEARCH METHOD

2.1 Threat Model and Security Scope

To critically assess the security posture of the proposed dual-cipher framework, we formulate a threat model under the assumption of the "honest-but-curious" Cloud Service Provider (CSP). In this model, the CSP can reliably store data and run protocols but might also seek to analyze stored data or communication logs to extract sensitive information.

2.1.1 Addressed Threats (In-Scope)

In relation to the threats that are being addressed within the scope of the system, we mainly use the defense approach of securing data-at-rest against compromise in the case of cloud storage. In this regard, it makes the information inaccessible even when an attacker or rogue insider has gained physical access to the storage drives; this is achieved through AES-256 encryption, such that files cannot be read without their corresponding key. At the same time, the system will be designed to counteract packet sniffing and eavesdropping during transmission to mitigate passive network attacks. RSA-2048 is also crucial to increase the level of security as the passive attacker would not be able to decrypt AES keys concealed in this traffic despite intercepting the communication between the two sides of the network. In addition, the symmetric AES key is not sent in plaintext to address the threat of key exposure. Rather, it is encapsulated in the RSA cipher and this practically removes the possibility of the actual key being intercepted during the exchange process.

2.1.2 Limitations (Out-of-Scope)

It is worth adding that this model assumes that the device (client-side) used by the user is secure. The malware in the machine of the user, inadvertent sharing of keys by the user or even physical theft of the machine are vulnerabilities beyond the control of this cryptographic solution. Our work is strictly committed to network data security and cloud storage.

2.2 System Architecture

Figure 1 specifies the high-level architecture which is structured into four functional modules:

1. Client Module: This module facilitates pre-processing operations, where the user initializes the secure workflow and establishes the session parameters.
2. Encryption Engine: This is the most basic processing unit which performs the dual-cipher logic. It combines the AES algorithm of data transformation at high speed and encapsulation of keys using RSA algorithm.

3. Key Management System (KMS): The KMS is focused on key lifecycle. It also deals with the RSA key pairs generation and the guarantee of the secure distribution mechanisms.
4. Cloud Storage Interface: This module controls a transport layer, which is to ensure the safe delivery of the encrypted data to the cloud. It guarantees the persistence of data throughout the upload process and also renders the removal of encrypted objects in a smooth and clear-cut way.

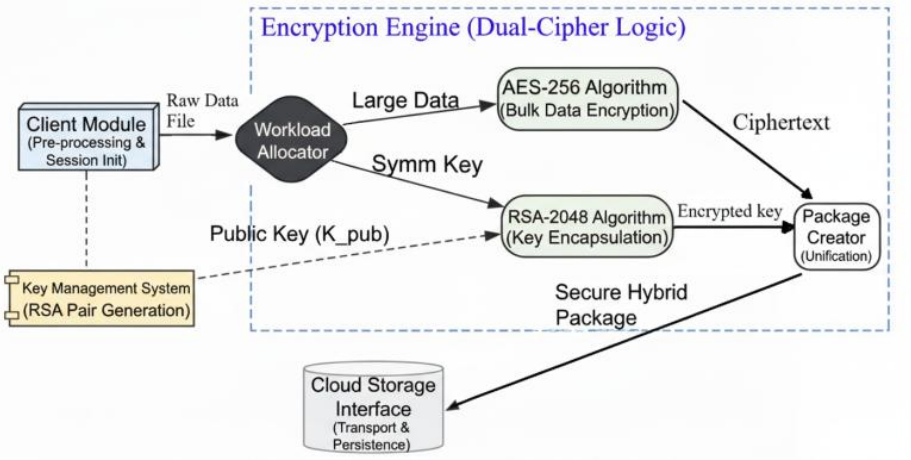

Fig. 1. Architectural framework of the proposed intelligent dual-cipher system illustrating the dynamic workload allocation.

2.3 Encryption and Decryption Algorithms

To ensure safe data-transformation and key delivery, the suggested system performs the encryption procedure in four systematized phases, the session start, Data encrypting operation with AES (AES Layer), Key encapsulating procedure with RSA (RSA Layer) and package generation. Algorithms 1 and 2 present the pseudo-code for encryption and decryption.

- Algorithm 1: Encryption Algorithm
  Function Dual_Cipher_Encrypt (plaintext, K_{pub}, F_Size)
  Phase 1: Initialization
  // Generate required components for symmetric encryption
  $K_{sym} \leftarrow PRNG(256)$ //Generate 256-bit AES Session Key (K_{sym})
  $IV \leftarrow PRNGen(128)$ //Initialization Vector
  $K_{pub} \leftarrow$ KMS. Retrieve_Public_Key //Retrieve K_{pub}
  Phase 2: Data Encryption with AES
  Encrypted_data = AES_Encrypt (plaintext, K_{sym}, IV, CBC_MODE)
  Phase 3: Key Encapsulation with RSA
  Encrypted_key = RSA_Encrypt (K_{sym}, K_{pub})
  Phase 4: Package for storage
  Cipher_Pkg= {Encrypted_Key, IV, Encrypted_Data}
  Cloud_Storage.Upload(Cipher_Pkg)
  Return cipher_pkg
  End Function
- Algorithm 2: Decryption Algorithm
  Function Dual_Cipher_Decrypt (Cipher_Pkg, K_{priv}, F_Size)
  Phase 1: Data Analytics and Feature Extraction
  Extract {Encrypted_Key, IV, Encrypted_Data} from Cipher_Pkg
  Phase 2: Key Decapsulation
  K_{sym} = RSA_Decrypt (Encrypted_Key, K_{priv})
  Phase 3: Data Decryption
  Plaintext = AES_Decrypt (Encrypted_Data, K_{sym}, IV, CBC_MODE)
  Return plaintext
  End Function

2.4 Experimental Environment

All experiments were performed on a workstation equipped with an Intel Core i7 processor (3.4 GHz), 16 GB RAM, running Windows 11 (64-bit). The system was implemented in Python using the PyCryptodome cryptographic library. All encryption and decryption were performed in a single threaded fashion to create a stable environment to measure

performance. To ensure data consistency and reproducibility, synthetic files ranging from 1 MB to 1 GB were generated using Python's os.urandom function to simulate high-entropy unstructured data. Each of the experiments was repeated five times to reduce the amount of random noise in the results.

2.5 Dataset Description

A test corpus of synthetic binary files with high entropy has been created to represent a variety of real-world workloads on clouds. The test corpus includes four orders of magnitude: 1- 100 MB, and 1 GB. Each of the experiments was repeated five times in order to obtain statistical reliability and remove the effects of momentary variations in the system.

3. RESULTS AND DISCUSSION

This part explains the empirical benchmarking of the provided Dual Cipher framework. The main goal of our project is to assess the computational cost of the system in relation to its security value in file sizes of between 1 MB and 1GB. For a strict comparative analysis, we benchmark the hybrid scheme against standalone AES-256 and RSA-2048 baselines under identical test conditions. In the ensuing subsections, key metrics namely execution latency, throughput, and CPU consumption are examined and finally the security posture of the system discussed.

3.1 Encryption and Decryption Time Comparison

Table II shows the relationship between the encryption time of files of varying size and the respective encryption algorithm. Based on the table, it is evident that AES-256 takes the shortest period to encrypt a large data set. Nevertheless, RSA-2048 is the slowest encryption algorithm in case of large data encryption because of its limited security features. The dual cipher system as shown in Table 2 contributes at most 20-21 percent to the overall workload of the original AES algorithm. Given that cloud computing environments involve the transfer of secure keys, this additional workload is important in ascertaining whether the pace of the environment is affected.

TABLE II. ENCRYPTION TIME (MS) VS. FILE SIZE FOR EACH ALGORITHM

File Size	AES-256	RSA-2048	Hybrid	Overhead
1 MB	12	1840	18	+50%
10 MB	118	18200	142	+20%
100 MB	1165	N/A*	1398	+20%
1 GB	11820	N/A*	14250	+21%

Note: (N/A*) Using RSA alone for huge files is impractical.

Dual-cipher system is not highly sensitive to the file size increase and as Figure 2 depicts, the system remains constant. Also, though the dual-cipher system is not faster than AES per se, it provides much higher protection without adding much computational overhead of the system.

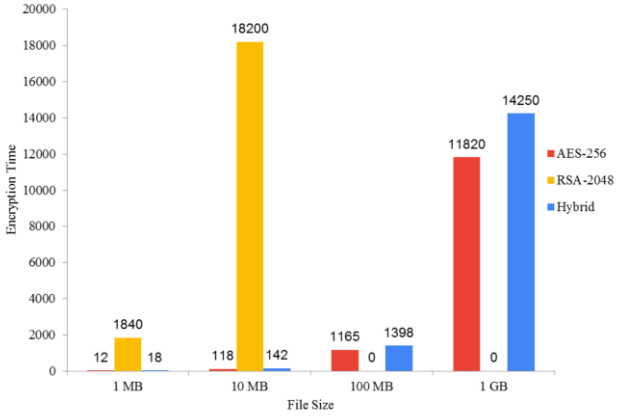

Fig. 2. Encryption time vs. file size

Table III shows a comparison between the three algorithms. the aes-256 and rsa-2048 algorithms, along with the hybrid aes-rsa algorithm, differ in decryption times for files of varying sizes. the first algorithm, aes-256, demonstrates faster decryption performance, averaging 10–11,650 ms across all file sizes, placing it among the top-ranked in terms of encryption speed. on the other hand, rsa-2048 is clearly slow even with small files, making it unsuitable for files larger than 10 mb. the hybrid algorithm aes-rsa achieves performance so close to that of aes, but with an additional overhead of only 20-21% for files larger than 10 mb, and about 60% for 1mb files. the additional cost is caused by using the rsa key-decryption step.

TABLE III. DECRYPTION TIME (MS) VS. FILE SIZE FOR EACH ALGORITHM

File Size	AES-256	RSA-2048	Hybrid	Overhead
1 MB	10	1825	16	+60%
10 MB	115	18150	138	+20%
100 MB	1150	N/A*	1380	+20%
1 GB	11650	N/A*	14050	+21%

Note: (N/A*) large datasets are computationally infeasible when encrypted using the direct RSA-2048 encryption. These entries are only stored to show the limits of scalability of the algorithm.

Figure 3 is a visualization of the decryption latency trends detailed in Table II. The graph shows a sharp difference in scalability: RSA-2048 shows an even at smaller file sizes, exponential processing time increase. On the contrary, the hybrid scheme traces the AES-256 base is similar, maintaining a stable and stable overhead with which it proves to be appropriate in large-scale data.

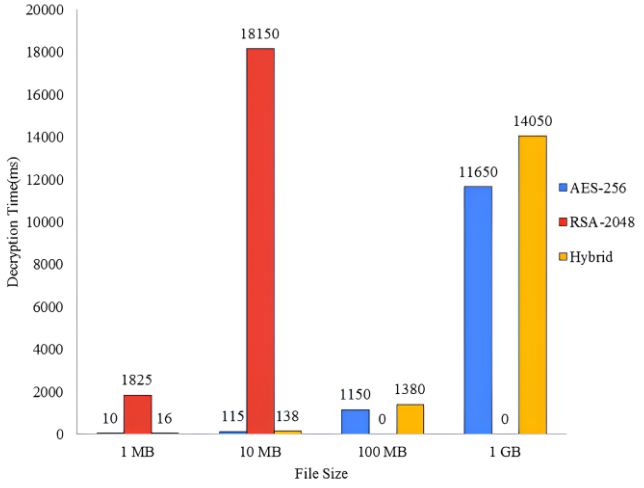

Fig. 3. Decryption time vs. file size

Figure 4 shows the time dissimilarities for each of the three algorithms in terms of encrtion and decryption times. The AES-256 algorithm is a little bit faster during decryption than encryption, by a margin of 5-10%, due to the necessities of the encryption process itself. The hybrid algorithm, on the other hand, boasts nearly identical speeds for both encryption and decryption (decryption being slightly faster). As for the RSA algorithm, its decryption performance of (1825 ms) is slightly quicker than its encryption performance, which goes (1840 ms) for files of one megabyte. This is due to private key operations are usually enhanced by using the Chinese Remainder Theorem. It is evident that using the hybrid algorithm provides almost identical speeds for encryption and decryption, which is necessary for practical cloud computing applications that require frequent and repeated access to data effectively.

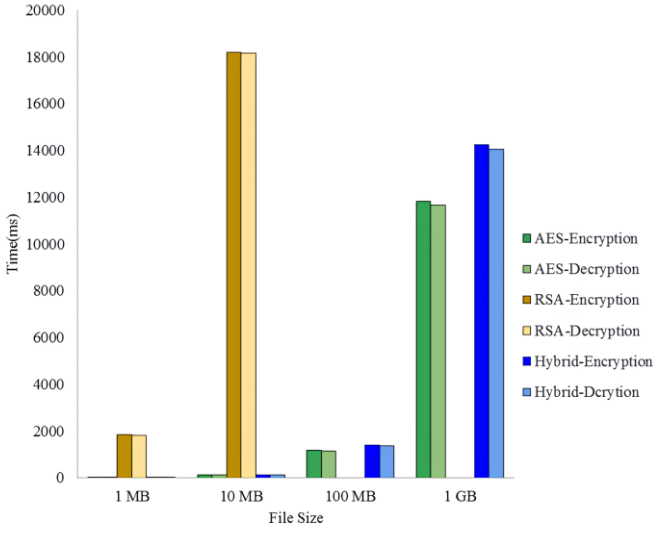

Fig. 4. Encryption vs. Decryption Time Comparison

3.2 Throughput and CPU Utilization

As Figure 5 shows, the maximum rate of throughput was obtained with the AES-256, with throughput of 83 MB /s to 86 MB/s with different file sizes. The hybrid algorithm was second having a more extensive performance range of about 56-72 MB/s. This is multiplying the number of bits/s that can be attributed to the extra computational load caused by the RSA encryption layer. Conversely, the throughput rate of the RSA-2048 algorithm was very low, not more than 0.5 MB/s, and thus not suitable when large files are to be encrypted.

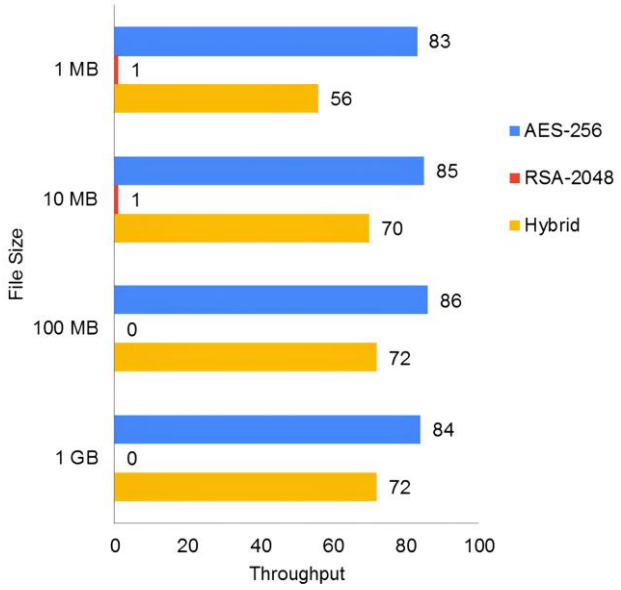

Fig. 5. Throughput comparison

Figure 6 demonstrates the performance and security of the algorithms is compared discloses that AES is very productive and less secure, and RSA is mediocre security with extremely. However, when the performance is low, such as displayed in this figure, the Hybrid AES-RSA proves to be the most appropriate to implement in cloud systems, since it is highly secured and performance oriented.

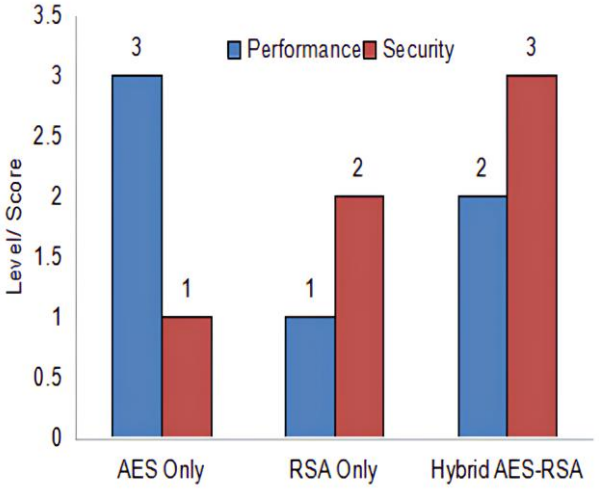

Fig. 6. Performance vs. security

Finally, Figure 7 demonstrates a linear dependence between the file size and the increased CPU use when used in the case of varying file sizes. AES-256, RSA-2048, and hybrid AES-RSA. In any algorithm, the bigger the file, the greater the CPU usage it requires.

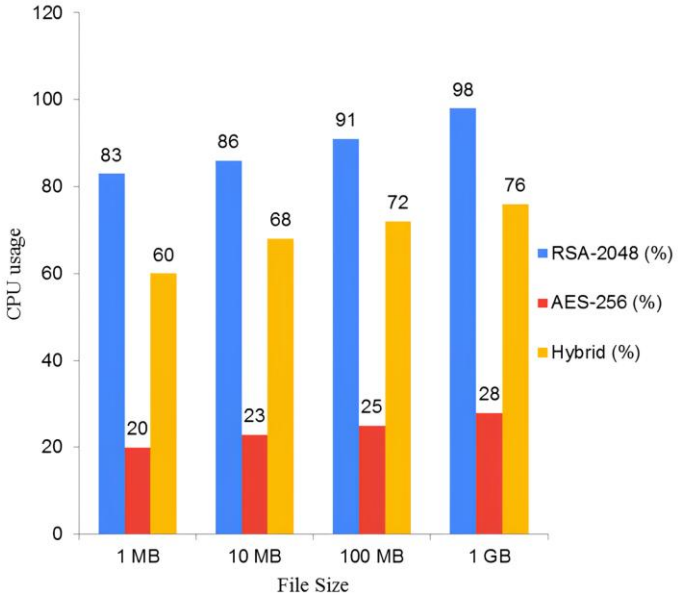

Fig. 7. CPU utilization during encryption

RSA was characterized by the largest consumption of the CPU, which was approximately 98% when it was processing large files. AES had the lowest CPU usage of between 20 and 28 % with different file sizes. The hybrid algorithm between AES-RSA had a balanced performance, with a range of 60-76% consumption of the CPU. This equilibrium is desirable in encrypting massive volumes of information with effective speed: AES manages encrypts bulk data in a short amount of time, and key exchange is handled by RSA. By integrating them, that will minimize the overall cost of computation without compromising security.

3.3 Security Strength Analysis.

Table IV exhibits the performance variation of AES-256, RSA-2048, and the hybrid AES-RSA system as regards to security and efficiency. AES-256 is faster in bulk data encryption but does not offer a secure medium of distributing keys. On the contrary, RSA- 2048 has strong key exchange algorithms, but it is computationally infeasible when used with large files.

TABLE IV. THE DIFFERENCE IN PERFORMANCE

Security Metric	AES-256	RSA-2048	Hybrid
Key Distribution	Weak	Strong	Strong
Bulk Encryption	Strong	Weak	Strong
Computational Security	2^256	2^112	2^112
Resistance to Attacks	Good	Good	Strong (bounded by RSA-2048 security level)
Forward Secrecy	No	Possible	Yes

The suggested two-hybrid model is based on a stacked security paradigm; but ultimately, its cryptographic resilience is mathematically constrained by the factorization difficulty of the RSA-2048 element. We clearly recognize the fact that though AES-256 has a larger theoretical level of security (2256), the actual security of the overall session is dictated by the 112-bit of RSA key encapsulation mechanism (KEM) security level. This tier is now defined as adequate and robust in accordance to NIST guidelines to secure commercial cloud applications. Any shift to more secure levels, e.g. RSA-3072 (128-bit security), would necessarily impose a disproportionate cost in terms of computing resources, breaking the real-time constraints of cloud storage. Accordingly, under an Architectural Decoupling strategy, our architecture balances its strategic side: it has a secure 112-bit threshold, with a stable and manageable performance overhead of about 21%. Additionally, such image-specific parameters as NPCR, UACI, or histogram analysis are deliberately excluded in the analysis of this framework because these parameters are statistically insignificant to all-purpose cloud objects and data that are independent of any multimedia. Rather, we evaluate basic cryptography properties, which include: Key Space Entropy, Forward Secrecy and Brute-force Attacks. These measures are final integrity markers of data integrity in cloud

systems. We compare our results with modern variants such as AES-ECC and AES-OTP-RSA and our results indicate that the suggested system can offer similar security assurances at a foreseeable performance cost (83-86% efficiency). Our design as compared to complex adaptive schemes which tend to compromise architectural stability in favor of fringe benefits espouses a philosophy of 'Secure Simplicity.' This guarantees an easy integration with the already available cloud service providers (CSPs) and a high sense of reproducibility. Finally, the experimental results validate the claim that the system based on dual ciphering provides a high security-performance scale which cannot be attained by the single RSA, especially when working with data sets of over 10MB.

4. CONCLUSION

In the current work, it was suggested that an intelligent cloud computing security approach rely on a dual hybrid cipher system that combines AES and RSA. The suggested model is effective in the balancing of high security and high computational efficiency using AES to encrypt bulk data and RSA to exchange keys in a secure manner. Experimental outcomes have shown that, on average, the overhead added by the hybrid approach is about 21% with AES alone being used on large files (>10 MB) with a significantly higher security and much better performance being obtained than with standalone RSA. All these empirical results justify the feasibility of framework to production grade cloud environment. The results show that a hybrid architecture will be able to implement high data security standards without becoming a bottleneck. Though such results are promising, it is important to note that there exist shortcomings in the experimental setup of the study. The validation was done mostly in single-threaded controlled simulation environment. These offer a robust framework on which to assess the efficiency of cryptography, but do not effectively represent the realities in the cloud environment, like resource contention between multiple tenants and unreliable network latency. Thus, the second step will be the deployment of such framework to distributed cloud systems (e.g., AWS or Azure) and test its scalability and resilience during the high-concurrency scenario.

Further development of the system will include enhancing the integrity and persistence layer of the system which will be done by integrating Hardware Security Modules (HSMs) to store keys in a manner that they cannot be tampered with. We are also intending to use such modes of authenticated encryption like AES-GCM to improve the authenticity of data. Further work will be done to implement the Post-Quantum Cryptographic (PQC) primitives to be resilient over the long term in the face of new computational threats entering the industry. In addition, we also plan to explore Trusted Execution Environments (TEEs) as part of confidential computing environments to protect privacy of data when it is measured by runtime execution.

Funding:

The authors confirm that no funding was acquired from any organization, grant agency, or institution. This research was undertaken without any external financial contributions.

Conflicts of Interest:

The authors declare no competing financial interests in this study.

Acknowledgment:

The authors would like to thank their institutions for providing the necessary facilities and guidance, which proved vital in achieving the study's objectives.

References

[1] A. Damaraju, "Cloud security challenges and solutions in the era of digital transformation," *International Journal of Advanced Engineering Technologies and Innovations*, vol. 1, no. 3, pp. 387–413, Jun. 2024. [Online]. Available: https://ijaeti.com/index.php/Journal/article/view/348

[2] S. M. Altowaijri and Y. El Touati, "Securing cloud computing services with an intelligent preventive approach," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 13998–14005, Jun. 2024, doi: 10.48084/etasr.7268.

[3] U. A. Butt *et al.*, "Cloud security threats and solutions: A survey," *Wireless Personal Communications*, vol. 128, no. 1, pp. 387–413, Jan. 2023, doi: 10.1007/s11277-022-09960-z.

[4] IBM Security, *Cost of a Data Breach Report 2024*. Armonk, NY, USA, 2024. [Online]. Available: https://www.ibm.com/reports/data-breach

[5] U. H. Shaikh *et al.*, "A comparative survey of symmetric and asymmetric key cryptography algorithms," in *Proc. 2nd Int. Multidisciplinary Conf. Emerging Trends in Engineering Technology (IMCEET)*, 2024, pp. 257–262. [Online]. Available: https://www.researchgate.net/publication/384663640

[6] S. Ahmad *et al.*, "Protecting data in the cloud: A systematic literature review of key management," *Concurrency and Computation: Practice and Experience*, vol. 37, no. 21–22, 2025, doi: 10.1002/cpe.70223.

[7] B. Halak, Y. Yilmaz, and D. Shiu, "Comparative analysis of energy costs of asymmetric vs symmetric encryption-based security applications," *IEEE Access*, vol. 10, pp. 76707–76719, 2022, doi: 10.1109/ACCESS.2022.3192970.

[8] M. S. Islam *et al.*, "Ensuring end-to-end IoT data security and privacy through cloud-enhanced confidential computing," in *[Book Chapter]*, 2024, pp. 71–91, doi: 10.1007/978-3-031-65172-45.

[9] M. Chauhan and S. Shiaeles, "An analysis of cloud security frameworks, problems and proposed solutions," *Network*, vol. 3, no. 3, pp. 422–450, Sep. 2023, doi: 10.3390/network3030018.
[10] Cloud Security Alliance, *Security Guidance for Critical Areas of Focus in Cloud Computing V5.0*, 2024. [Online]. Available: https://cloudsecurityalliance.org/research/guidance#5
[11] K. Harrington, "Shared responsibility model in cloud security," ResearchGate. [Online]. Available: https://www.researchgate.net/publication/392064822
[12] H. Harvin, *Handbook of Cryptography*. Henry Harvin, 2023. [Online]. Available: https://books.google.com/books?id=8OraEAAAQBAJ
[13] L. E. Hughes, "Basic cryptography: Symmetric key encryption," in *Pro Active Directory Certificate Services*. Berkeley, CA, USA: Apress, 2022, pp. 3–17, doi: 10.1007/978-1-4842-7486-6_1.
[14] R. K. Muhammed *et al.*, "Comparative analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for image encryption," *Kurdistan Journal of Applied Research*, vol. 9, no. 1, pp. 52–65, May 2024, doi: 10.24017/science.2024.1.5.
[15] Z. A. Mohammed *et al.*, "Advancing cloud image security via AES algorithm enhancement techniques," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12694–12701, Feb. 2024, doi: 10.48084/etasr.6601.
[16] S. Dey, G. Leander, and N. K. Sharma, "Improved key recovery attacks on reduced-round Salsa20," *Designs, Codes and Cryptography*, vol. 93, no. 1, pp. 243–262, Jan. 2025, doi: 10.1007/s10623-024-01522-7.
[17] T. Beyne, Y. L. Chen, and M. Verbauwhede, "A robust variant of ChaCha20-Poly1305," 2025. [Online]. Available: https://eprint.iacr.org/2025/222
[18] K. Sasikumar and S. Nagarajan, "Comprehensive review and analysis of cryptography techniques in cloud computing," *IEEE Access*, vol. 12, pp. 52325–52351, 2024, doi: 10.1109/ACCESS.2024.3385449.
[19] Divya and S. Upasna, "Evaluating RSA key length: Impact on security hardness and computational efficiency," *JOIREM*, vol. 10, no. 8, pp. 1–6, Aug. 2024. [Online]. Available: https://joirem.com
[20] M. I. Mihailescu and S. L. Nita, "Elliptic-curve cryptography," in *Pro Cryptography and Cryptanalysis with C++23*. Berkeley, CA, USA: Apress, 2023, pp. 207–243, doi: 10.1007/978-1-4842-9450-5_9.
[21] J. Bharti and S. Singh, "Securing cloud data: A comprehensive analysis of encryption techniques and recent advancements," in *Computational Intelligence and Mathematical Applications*. London, U.K.: CRC Press, 2024, pp. 143–148, doi: 10.1201/9781003534112-23.
[22] S. Kumar and D. Kumar, "Securing cloud storage data using hybrid AES-ECC cryptographic approach," *Journal of Mobile Multimedia*, Nov. 2022, doi: 10.13052/jmm1550-4646.1921.
[23] D. Shivaramakrishna and M. Nagaratna, "A novel hybrid cryptographic framework for secure data storage in cloud computing," *Alexandria Engineering Journal*, vol. 84, pp. 275–284, Dec. 2023, doi: 10.1016/j.aej.2023.10.054.
[24] L. Zhao *et al.*, "Data compression and encryption fusion: A review of hybrid techniques for secure and efficient online transmission," *IEEE Access*, vol. 13, pp. 98791–98805, 2025, doi: 10.1109/ACCESS.2025.3575428.