WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Simulation Based Performance Analysis of Hybrid Advanced Encryption Standard, Data Encryption Standard and Rivest Shamir Adleman Algorithms

Ashiru Musa[1], *Hassan Buhari Mamman*[2] , *Hassan Sabo Miya*[3], *Ibrahim Salim Yalwa*[4], *Murtala Aminu Baba*[5]

[1]Department of Electrical and Electronics Engineering, Abubakar Tafawa Balewa University, Nigeria
[2]Department of Electrical and Electronics Engineering, Abubakar Tafawa Balewa University, Nigeria
[3]Department of Mechatronics and System Engineering, Abubakar Tafawa Balewa University, Nigeria
[4] Department of Electrical and Electronics Engineering, Abubakar Tafawa Balewa University, Nigeria
[5]Department of Computer and Communications Engineering, Abubakar Tafawa Balewa University, Nigeria

ABSTRACT:

The increasing demand for secure and efficient data communication has intensified the need for cryptographic systems that balance robustness with computational performance. Traditional encryption algorithms such as AES, DES, and RSA each exhibit notable limitations AES incurs higher computational overhead for large data sizes, DES faces security weaknesses and reduced efficiency, while RSA suffers from significant processing delays due to its complex mathematical operations. This study proposes and evaluates a Hybrid AES–DES–RSA encryption model designed to leverage the strength of symmetric and asymmetric techniques while mitigating their individual weaknesses. The hybrid system applies DES and AES for fast bulk data encryption and utilizes RSA solely for secure session key management. Performance evaluation was conducted through Python simulations across various file sizes ranging from 32 KB to 1024 KB, with metrics including memory usage, and throughput. Results show that the hybrid model consistently outperforms the standalone algorithms, achieving lower memory consumption, and significantly higher throughput. The system provides average performance improvements of *9.37% over AES*, *33.33% over DES*, and *59.03% over RSA*, demonstrating its scalability and operational efficiency. The findings confirm the hybrid model as a viable solution for secure, resource-efficient data communication suitable for applications in cloud computing, IoT environments, and real-time systems.

Keywords: Hybrid Encryption; AES; DES; RSA; Cryptography; Data Security; Performance Analysis.

1. INTRODUCTION

1.1 Background of the Study

In our current digitalized Environment, the protection of information has become one of the most critical challenges in computing and communication systems. With the exponential growth of data exchange across cloud networks, mobile devices, and IoT infrastructures, ensuring confidentiality, integrity, and authenticity of transmitted data is paramount. Cryptography plays a fundamental role in achieving this goal by transforming plaintext into ciphertext, thereby making information unintelligible to unauthorized users (Banerjee, 2024)

Modern cryptographic systems are primarily classified into symmetric and asymmetric algorithms. Symmetric encryption, such as the Advanced Encryption Standard (AES) and Data Encryption Standard (DES), uses a single shared key for both encryption and decryption(Rizk-Allah et al., 2023). These algorithms are well-known for their speed and computational efficiency, which makes them appropriate for protecting massive amounts of data.

(Laia et al., 2021). However, symmetric systems face the challenge of secure key distribution. In contrast, asymmetric algorithms such as Rivest–Shamir–Adleman (RSA) use a pair of public and private keys, providing enhanced security and scalability in open communication environments (Zhang et al., 2019). Despite this, RSA's computational complexity results in slower encryption and decryption times, particularly for large datasets

(Banerjee, 2024).

The Data Encryption Standard (DES), developed in the 1970s, was one of the earliest block cipher algorithms used for secure data transmission. Although it provided a foundation for modern encryption, its 56-bit key length becomes subject to brute-force assaults as its computational capacity advanced (Sood & Kaur, 2023). To mitigate these weaknesses, Triple DES (3DES) was introduced, increasing security by applying the DES algorithm three times. However, 3DES also increased computational overhead, making it unsuitable for high-speed applications (Sood & Kaur, 2023).

The Advanced Encryption Standard (AES) employs substitution–permutation networks and supports key sizes of 128, 192, and 256 bits, offering a strong defence against brute-force and differential attacks (Kadhim & Kamal, 2018; Sarkar et al., 2024). However, AES is still subject to side-channel attacks, like timing or power analysis attacks, and its complex key scheduling can impose a computational burden in resource-constrained environments (Ramakrishna & Ali Shaik, 2025).

On the other hand, RSA provides robust security through its reliance on the computational difficulty of large prime factorization. It is widely used for digital signatures, secure key exchange, and authentication in hybrid systems (Abroshan, 2021). Nevertheless, RSA's performance degrades when processing large data volumes due to its intensive mathematical computations and long key lengths, which lead to increased encryption and decryption time (Banerjee, 2024).

Given these limitations, researchers have turned to hybrid encryption models that combine symmetric and asymmetric techniques to achieve both high security and efficiency. In such systems, symmetric algorithms like AES or DES handle bulk data encryption, while asymmetric algorithms like RSA manage secure key exchange. (Zhang et al., 2019) proposed a hybrid encryption framework combining an improved AES (P-AES) with RSA for medical data storage in cloud databases, achieving faster processing speed and higher security. Similarly, Dynamic AES and Blockchain-based key management systems have been proposed to enhance security, reduce key compromise risks, and improve scalability in cloud environments (Shakor et al., 2024).

Recent research trends emphasize simulation-based performance evaluation to determine the optimal combination of algorithms for balancing encryption speed, memory usage, and security (Pothireddy et al., 2024). Hybrid models not only mitigate the weaknesses of individual algorithms but also offer flexible scalability for various applications such as cloud computing, IoT, and e-health systems (Pothireddy et al., 2024).

Therefore, developing and analysing a hybrid encryption model integrating AES, DES, and RSA is crucial to address the current challenges in encryption performance and security (Pothireddy et al., 2024). Such a model can leverage the computational efficiency of symmetric algorithms, the robust key management of asymmetric algorithms, and the enhanced throughput achieved through simulation-based optimization(Muttaqin & Rahmadoni, 2020). This study thus contributes to the ongoing effort to design secure, efficient, and reliable encryption frameworks suitable for modern digital communication systems.

1.2 Problem Statement

Existing encryption algorithms face significant challenges: AES has key distribution problem, DES is weak against brute-force attacks, and RSA is slow. Additionally, all three are vulnerable to side-channel attacks. To enhance security and efficiency, a hybrid encryption approach is needed to combine their strengths while mitigating their weaknesses.

2. MATERIALS AND METHODS

2.1 Materials and Tools

The following are the equipment needed for this research:

- Laptop computer
- Python with Libraries

2.2 Methods

The methodology adopted in this research is the algorithm development of the individual algorithms (AES, DES, and RSA) were implemented in Python using the PyCryptodome library. Each algorithm was tested independently to ensure accuracy and correctness before integration into the hybrid model. ***2.2.1 Procedure for gathering data***

The data collection procedure for this research involves the acquisition of plaintext data samples of varying sizes to test encryption and decryption efficiency. No sensitive or personal data were used; instead, generic text files of 32 KB, 64 KB, 128 KB, 512 KB and 1024 KB were generated using random text generators.

Parameter Recording: For each run, the following data was collected:

- Memory usage (in megabytes)
- Throughput (in bytes per second)

2.3 Algorithm Implementation

The hybrid encryption system was implemented in Python to simulate and analyze its performance. The model combines AES, DES, and RSA to achieve optimal balance between speed, security, and resource efficiency. The implementation follows these key steps:

1. **Step 1 – Key Generation:**

A random **DES key** and **AES key** are generated for the symmetric encryption.

RSA generates a **public–private key pair** for asymmetric encryption.

2. **Step 2 – DES Encryption:**

The plaintext P is encrypted using DES:

$$C_1 = E_{K_{DES}}(P) \qquad (1)$$

3. **Step 3 – AES Encryption:**

The DES output $C_1$is re-encrypted using AES:

$$C_2 = E_{K_{AES}}(C_1) \qquad (2)$$

4. **Step 4 – RSA Key Encryption:**

The AES key is encrypted using the RSA public key:

$$K'_{AES} = E_{K_{RSA,pub}}(K_{AES}) \qquad (3)$$

5. **Step 5 – Transmission:**

The encrypted data $C_2$and encrypted key $K'_{AES}$are transmitted to the receiver.

6. **Step 6 – Decryption at Receiver Side:**

The receiver uses the RSA private key to recover the AES key:

$$K_{AES} = D_{K_{RSA,priv}}(K'_{AES}) \quad (4)$$

AES decrypts $C_2$to retrieve C_1:

$$C_1 = D_{K_{AES}}(C_2) \quad (5)$$

DES decrypts $C_1$to obtain the original plaintext P:

$$P = D_{K_{DES}}(C_1) \quad (6)$$

7. **Step 7 – Performance Logging:**

Execution time, memory usage, and throughput are logged automatically using Python scripts.

The figure shows the diagram of the proposed hybrid model.

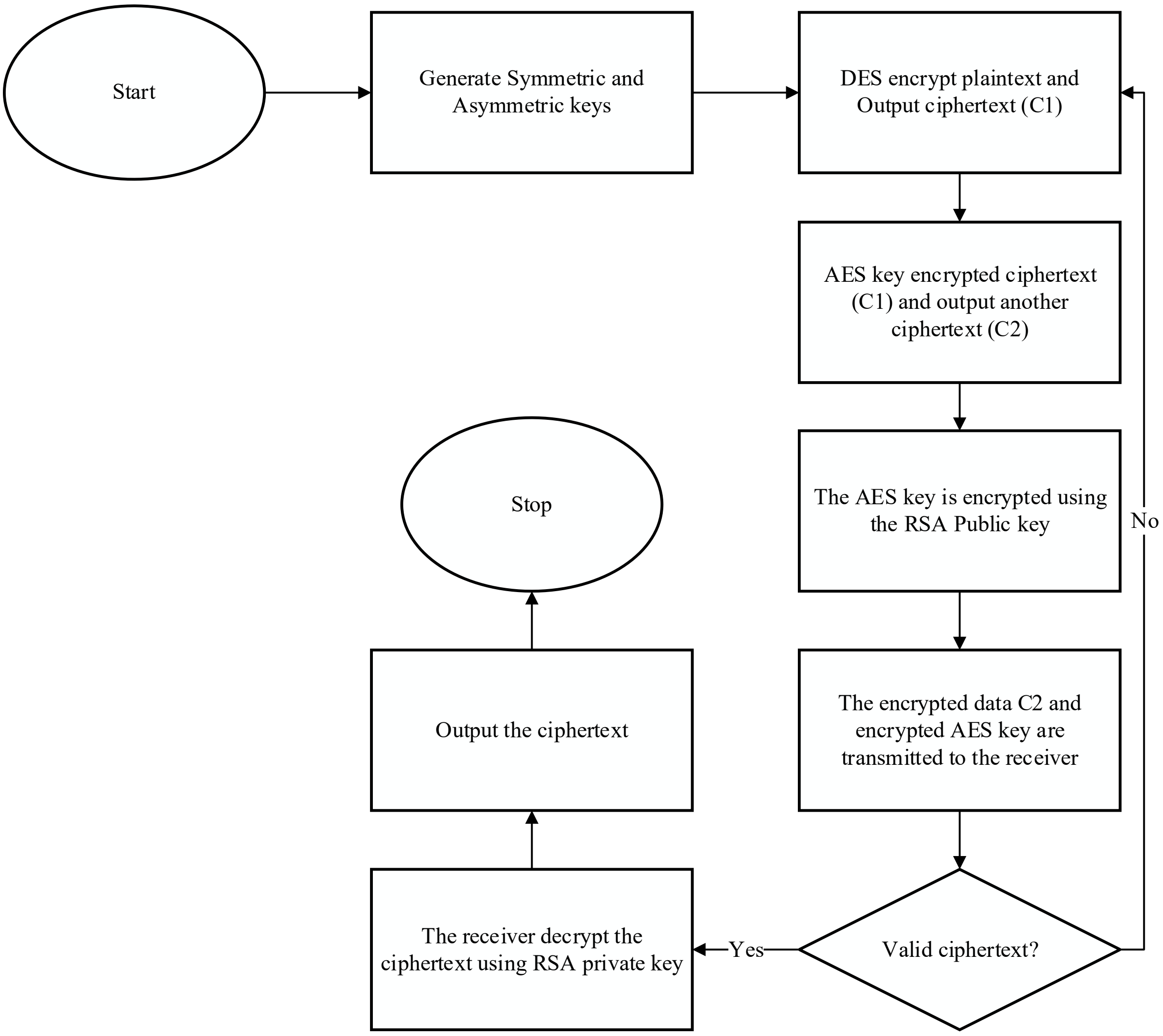
Start
Generate Symmetric and Asymmetric keys
DES encrypt plaintext and Output ciphertext (C1)
AES key encrypted ciphertext (C1) and output another ciphertext (C2)
The AES key is encrypted using the RSA Public key
The encrypted data C2 and encrypted AES key are transmitted to the receiver
Valid ciphertext?
No
Yes
The receiver decrypt the ciphertext using RSA private key
Output the ciphertext
Stop

Fig. 1- Flowchart of the proposed hybrid model

3. Performance Evaluation Metrics

E The performance of AES, DES, RSA and Hybrid model are evaluated based on the following key parameters:

1. **Memory Usage (M_{used}):**

The difference between peak and initial memory consumption during execution in megabytes (MB).

$M_{used} = M_{peak} - M_{base}$ (7)

$Where\ M_{peak}\ is\ the\ peak\ memory\ and\ M_{base}\ is\ the\ base\ or\ initial\ memory$

2. **Throughput (BPS):**

   The rate of data successfully processed per unit time in bits or bytes per seconds.

$Throughput = \frac{D}{T}$ (8)

$Where\ D\ is\ the\ total\ data\ encrypted\ in\ bits\ or\ bytes, and\ T\ is\ the\ total\ time\ taken\ in\ seconds.$

4. Results

4.1 Introduction

This section displays and analyzes the findings from the simulation-based performance analysis of the AES, DES, RSA, and the proposed Hybrid AES–DES–RSA encryption model. The evaluation focuses on four key performance metrics: memory usage, and throughput across different file sizes (32 KB to 1024 KB). The findings from the four classical algorithms are compared with the performance of the hybrid system to determine efficiency, scalability, and suitability for secure data communication.

4.2 Memory Usage Analysis

Memory usage quantifies the average RAM consumed during encryption and decryption.

The table below summarizes the memory usage in megabytes (MB) for the algorithms.

Table 1: Memory Usage Comparison

File Size (KB)	AES (MB)	DES (MB)	RSA (MB)	Hybrid (MB)
32	0.71119	0.73438	0.71328	0.67578
64	0.25234	0.26216	0.26797	0.26172
128	0.47891	0.4525	0.53438	0.44141
256	0.7881	0.83281	0.78438	0.76953
512	1.51562	1.51953	1.50000	1.27344
1024	2.08297	2.09172	3.28125	2.07031

The figure below illustrates the Memory Usage for each algorithm across the different file sizes.

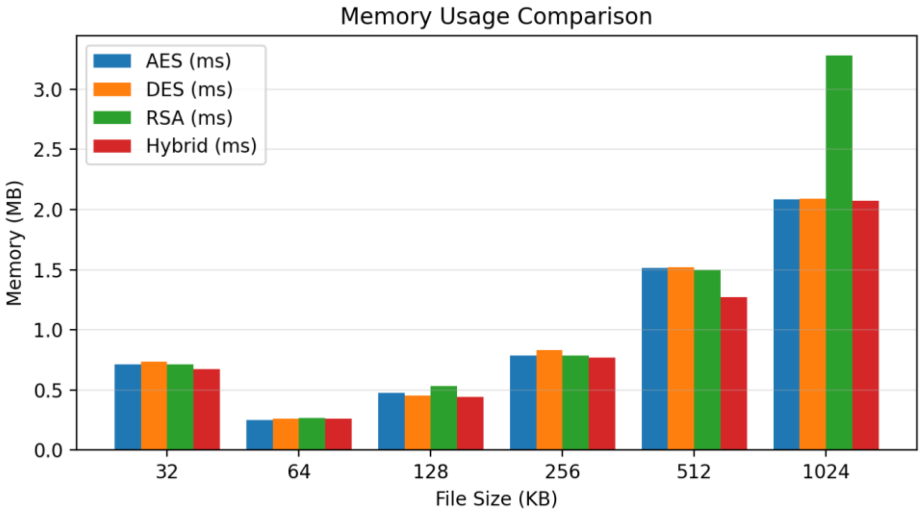

Fig. 2- Memory Usage Comparison

Results show:

RSA uses the highest memory, averaging above 1.5 MB for large files and over 3.2 MB at 1024 KB. This is due to large integer operations and key sizes.

AES and DES show moderate memory usage, with AES being slightly higher because of S-box tables.

The hybrid model has the lowest memory consumption across all file sizes. This is because it uses lightweight symmetric operations (AES and DES) for data, while RSA operates only on small key values.

The hybrid model is memory-efficient and well-suited for resource-constrained systems such as IoT or embedded applications.

4.3 Throughput Comparison

Throughput is a crucial metric that indicates the efficiency of the encryption and decryption processes, measured in bytes per second (Bps). The following table summarizes the throughput for the algorithms across different file sizes.

Table 2: Throughput Comparison

File Size (KB)	AES (Bps)	DES (Bps)	RSA (Bps)	Hybrid (Bps)
32	150.390	138.160	107.200	150.770
64	600.500	278.500	211.700	613.610
128	491.600	364.900	121.730	575.050
256	600.700	402.200	173.270	655.270
512	1800.600	1700.200	1240.200	1909.190
1024	4006.840	2564.100	1234.940	5224.890

The Figure below illustrates the throughput for each algorithm across the different file sizes.

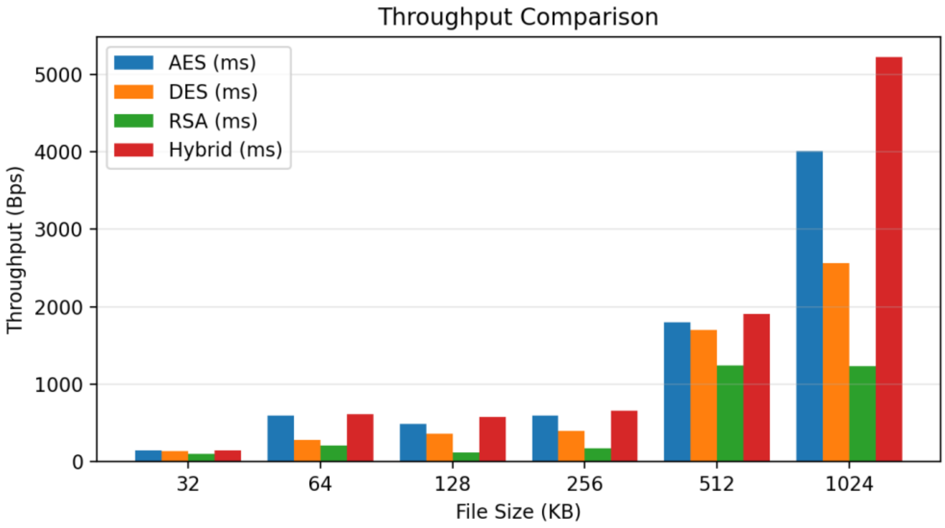

Fig. 3- Throughput Comparison

The hybrid model consistently achieves the highest throughput across all file sizes. For instance: At 256 KB, Hybrid: 655.270 Bps (highest) and At 1024 KB, Hybrid: 5224.89 Bps (highest overall)

This superior performance is attributed to: Reduced RSA workload (key-only encryption), Parallel or sequential symmetric operations (DES, AES) and The minimized encryption time results in maximized throughput

RSA has the lowest throughput, reflecting its slow encryption times and computational burden. AES and DES exhibit moderate throughput consistent with their symmetric efficiency.

Hybrid encryption is the most efficient model, capable of handling large data at high speed.

4.4 Percentage Improvement Comparison

Table 5 quantifies the performance gains of the hybrid model in terms of throughput improvement compared to the individual algorithms. The hybrid model shows progressive improvement, especially at larger file sizes.

Table 3: % Improvement by throughput

File Size (KB)	AES (%)	DES (%)	RSA (%)
32	0.252	8.364	28.898
64	2.137	54.613	65.564
128	14.512	36.545	78.831
256	8.328	38.621	73.558
512	5.688	10.947	35.041
1024	23.313	50.925	76.384

The figure below illustrates the % improvement of the Hybrid by throughput for each algorithm across the different file sizes.

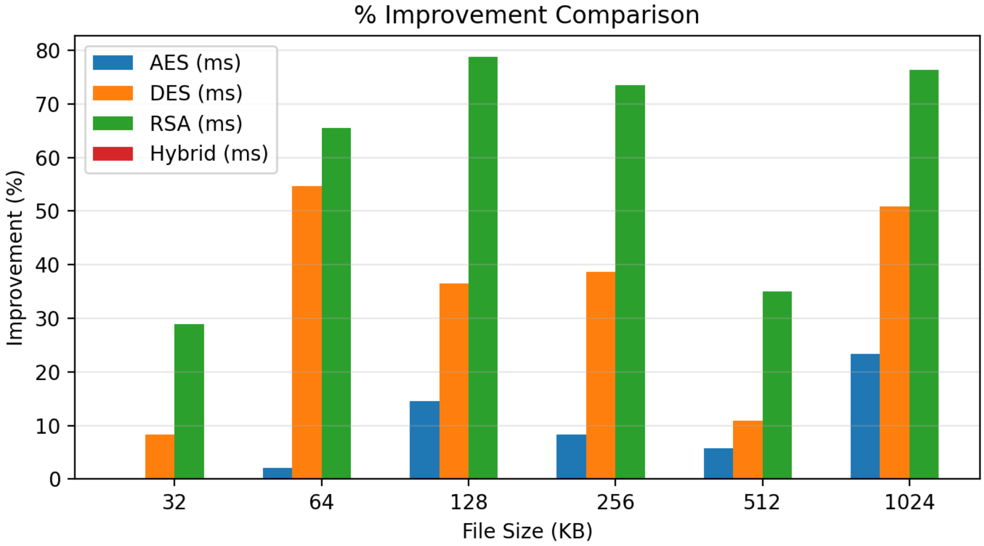

Fig. 4- % Improvement Comparison

Improvement over AES

The hybrid model improves AES by 0.25% to 23.31% depending on file size.

At small sizes (32–64 KB), AES already performs well, so improvement is small. At larger sizes (512–1024 KB), hybrid performance increases significantly. The highest improvement (23.31% at 1024 KB) indicates that the hybrid model scales better for large data volumes.

Average Improvement over AES = $\frac{60.25+2.14+14.51+8.32+5.69+23.31}{6} = 9.3\%$

Hybrid improves AES performance by an average of 9.37%.

Improvement over DES

The hybrid model improves DES by 8.36% to 54.61% depending on file size.

DES is slower and less efficient than AES. The hybrid combines DES and AES, making DES's performance vastly improved. The highest improvement (54.61% at 64 KB) confirms that the hybrid model significantly reduces the computational waste of DES.

Average Improvement over DES = $\frac{\mathbf{8.36+54.61+36.54+38.62+10.94+50.93}}{6} = 33.33\%$

Hybrid improves DES performance by an average of 33.33%.

Improvement over RSA

The hybrid model shows the greatest gains over RSA: 28.89% to 78.83%.

RSA is extremely slow for large data because of its modular exponentiation. The hybrid solution reduces RSA usage to only encrypting session keys. This dramatically reduces time and memory consumption, leading to performance boosts of over 70% for larger file sizes.

Average Improvement over RSA = $\frac{628.89+65.49+78.83+73.56+35.04+76.38}{6} = 59.03\%$

Hybrid improves RSA performance by an average of 59.03%.

5.0 CONCLUSIONS

From the results, several important conclusions can be drawn:

1. The study successfully developed and evaluated a Hybrid AES–DES–RSA encryption model aimed at improving the performance and security of data encryption systems.
2. The hybrid approach combined the strengths of symmetric algorithms (AES and DES) with the robust key management capabilities of RSA, while minimizing their individual limitations. Simulation results showed that the hybrid model significantly outperformed AES, DES, and RSA in terms of memory usage, and throughput.

5.1 Contributions to Knowledge

This research provides several significant contributions to the field of cryptography and secure communication:

1. Development of an Optimized Hybrid Encryption Model:

   A novel AES–DES–RSA hybrid encryption framework was designed to combine speed, efficiency, and secure key management, offering a new approach to balancing security and performance.
2. Quantified Performance Improvements:

   The study provides empirical evidence of performance gains achieved by the hybrid system:

   - 9.37% improvement over AES
   - 33.33% improvement over DES
   - 59.03% improvement over RSA

These results contribute new comparative metrics to the existing literature on hybrid cryptography.

3. Efficient Resource Utilization:

   The hybrid model demonstrated the lowest memory usage among the compared algorithms, showing that hybrid systems can be optimized for resource-constrained environments.

Throughput-Driven Cryptographic Optimization:
The research shows that combining symmetric and asymmetric encryption techniques can increase throughput, making encryption suitable for high-speed data environments such as cloud computing and multimedia transmission.

REFERENCES

Abroshan, H. (2021). A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and Asymmetric Cryptography Algorithms. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 12, Issue 6). www.ijacsa.thesai.org

Banerjee, S. (2024). Exploring Cryptographic Algorithms: Techniques, Applications, and Innovations. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal*, *4*(1), 607–620. https://doi.org/10.48175/ijarsct-18097ï

Kadhim, A. F., & Kamal, Z. A. (2018). Generating dynamic S-BOX based on Particle Swarm Optimization and Chaos Theory for AES. *Iraqi Journal of Science*, *59*(3), 1733–1745. https://doi.org/10.24996/IJS.2018.59.3C.18

Laia, O., Zamzami, E. M., & Sutarman. (2021). Analysis of Combination Algorithm Data Encryption Standard (DES) and Blum-Blum-Shub (BBS). *Journal of Physics: Conference Series*, *1898*(1). https://doi.org/10.1088/1742-6596/1898/1/012017

Muttaqin, K., & Rahmadoni, J. (2020). Analysis and Design of File Security System Aes (Advanced Encryption Standard) Cryptography Based. *Journal of Applied Engineering and Technological Science*, *1*(2), 113–123. https://doi.org/10.37385/jaets.v1i2.78

Pothireddy, S., Peddisetty, N., Yellamma, P., Botta, G., & Gottipati, K. N. (2024). Data Security in Cloud Environment by Using Hybrid Encryption Technique: A Comprehensive Study on Enhancing Confidentiality and Reliability. *International Journal of Intelligent Engineering and Systems*, *17*(2), 159–170. https://doi.org/10.22266/ijies2024.0430.14

Ramakrishna, D., & Ali Shaik, M. (2025). A Comprehensive Analysis of Cryptographic Algorithms: Evaluating Security, Efficiency, and Future Challenges. *IEEE Access*, *13*, 11576–11593. https://doi.org/10.1109/ACCESS.2024.3518533

Rizk-Allah, R. M., Abdulkader, H., Elatif, S. S. A., Oliva, D., Sosa-Gómez, G., & Snášel, V. (2023). On the Cryptanalysis of a Simplified AES Using a Hybrid Binary Grey Wolf Optimization. *Mathematics*, *11*(18). https://doi.org/10.3390/math11183982

Sarkar, B., Saha, A., Dutta, D., De Sarkar, G., & Karmakar, K. (2024). A Survey on the Advanced Encryption Standard (AES): A Pillar of Modern Cryptography. *International Journal of Computer Science and Mobile Computing*, *13*(4), 68–87. https://doi.org/10.47760/ijcsmc.2024.v13i04.008

Shakor, M. Y., Khaleel, M. I., Safran, M., Alfarhood, S., & Zhu, M. (2024). Dynamic AES Encryption and Blockchain Key Management: A Novel Solution for Cloud Data Security. *IEEE Access*, *12*, 26334–26343. https://doi.org/10.1109/ACCESS.2024.3351119

Sood, R., & Kaur, H. (2023). A Literature Review on RSA, DES and AES Encryption Algorithms. In *Emerging Trends in Engineering and Management* (pp. 57–63). Soft Computing Research Society. https://doi.org/10.56155/978-81-955020-3-5-07

Zhang, F., Chen, Y., Meng, W., & Wu, Q. (2019). HYBRID ENCRYPTION ALGORITHMS FOR MEDICAL DATA STORAGE SECURITY IN CLOUD DATABASE. *International Journal of Database Management Systems*, *11*(01), 57–73. https://doi.org/10.5121/ijdms.2019.11104