iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81197

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Analysis of AES and RSA Encryption Techniques for Secure Data Transmission

Vishwanath Irappa Baggol[1], Dr. H R Bhargava[2]
[1]*Student,* [2]*Professor, Department of forensic Science, Garden City University Bangalore, India 560049*

Abstract: ***In today's digital world, large amounts of sensitive information are transmitted across networks, making data security a critical requirement. Protecting this information from unauthorised access has become increasingly important, especially in applications such as online transactions, cloud storage, and secure communications. Cryptographic techniques are widely used to safeguard data by converting it into an unreadable format that can only be accessed by authorized users.***

This study presents a practical comparison of two commonly used encryption algorithms: the Advanced Encryption Standard (AES) and the Rivest Shamir Adleman (RSA) algorithm. AES is a symmetric encryption method known for its fast-processing speed and efficiency, whereas RSA is an asymmetric algorithm mainly used for secure key exchange and authentication.

The experiment was carried out using the OpenSSL cryptographic tool in a Windows PowerShell environment. A set of files ranging from approximately 1 KB to 21 KB was used to analyse performance based on encryption time, decryption time, and throughput. Since RSA is not suitable for encrypting large files directly, a hybrid approach was adopted in which AES was used to encrypt the data, while RSA was used to secure the AES encryption key.

The results obtained from the experiment show that AES performs significantly faster and achieves higher throughput compared to RSA operations. Although RSA requires more computational effort, it plays an important role in ensuring secure key distribution. The study concludes that combining AES and RSA in a hybrid model provides an effective balance between performance and security for modern data transmission systems.

Keywords: AES, RSA, Symmetric Encryption, Asymmetric Encryption, Cryptographic framework, OpenSSL, Data Security, Throughput Analysis

I. INTRODUCTION

A. Introduction

Cryptography refers to the practice of protecting information by converting readable data into an encoded format that prevents unauthorised access. In digital communication systems, encryption is used to transform plaintext into ciphertext so that only authorised users possessing the correct key can retrieve the original data. Cryptographic techniques ensure confidentiality, integrity, and secure transmission of information across computer networks. These mechanisms are widely used in modern applications such as online banking, cloud computing, digital communication systems, and secure data storage, where sensitive information must be protected from interception or misuse (Pethe and Pande, 2017).

Encryption algorithms used in cryptographic systems are generally classified into two main categories: symmetric encryption and asymmetric encryption. Symmetric encryption uses a single secret key for both encryption and decryption processes. Because the same key is shared between communicating parties, this method provides faster processing speed and is efficient for encrypting large volumes of data. Asymmetric encryption, on the other hand, uses a pair of mathematically related keys known as the public key and private key. The public key is used for encryption while the private key is used for decryption, enabling secure key exchange and authentication in communication systems (Verma and Sharma, 2020).

Among the symmetric encryption techniques, the Advanced Encryption Standard (AES) has become one of the most widely used algorithms due to its strong security structure and efficient performance. AES operates on fixed-size data blocks and supports multiple key lengths, which makes it suitable for various security applications. In contrast, the Rivest–Shamir–Adleman (RSA) algorithm represents a commonly used asymmetric encryption technique that relies on mathematical operations involving large prime numbers to provide secure communication. While AES is efficient for encrypting bulk data, RSA is primarily used for secure key exchange and digital signatures (Sholikhatin *et al*., 2022).

Modern secure communication systems often combine both symmetric and asymmetric encryption methods in a hybrid model. In such systems, symmetric encryption algorithms like AES are used to encrypt the actual data, while asymmetric algorithms such as RSA are used to securely exchange the symmetric encryption key. This approach allows systems to achieve both computational efficiency and strong security in practical implementations **(Akter *et al.*, 2023).**

B. Research Scope

This study concentrates on experimentally analysing the performance of AES and RSA encryption techniques using the OpenSSL cryptographic library. Testing is conducted within a Windows PowerShell environment to measure real execution time and throughput values. Files ranging from 1 KB to approximately 21 KB are selected to observe performance behaviour across varying data sizes.

Primary emphasis is placed on encryption time, decryption time, and throughput calculations. Because RSA cannot directly encrypt larger files due to inherent size restrictions, a hybrid encryption approach is implemented. Under this model, AES encrypts the data file, while RSA secures the AES session key. The investigation is limited to performance benchmarking and does not include theoretical cryptanalysis or attack modelling.

C. Rationale of the Study

Growing digital dependence demands encryption methods that provide both security strength and operational efficiency. Differences in algorithm structure result in noticeable variations in processing speed and computational load between symmetric and asymmetric techniques. While symmetric algorithms are generally faster, asymmetric methods involve mathematically intensive operations that increase execution time.

Existing comparisons often focus on theoretical explanations rather than hands-on experimentation using widely deployed cryptographic tools. Practical benchmarking using OpenSSL offers a clearer understanding of how these algorithms perform in actual implementation environments. Conducting structured performance measurements allows identification of strengths and limitations in measurable terms.

Through systematic experimentation and analysis, this research contributes to informed decision-making regarding algorithm selection for secure data transmission in modern computing systems.

II. REVIEW OF LITERATURE

A. Review of literature

The increasing reliance on digital communication systems has made information security a critical research area. Encryption techniques are essential to protect data from unauthorised access during transmission and storage. Cryptographic algorithms are broadly classified into symmetric and asymmetric encryption methods, each offering different advantages in terms of performance and security.

- Manikandaprabhu and Samreetha (2024) reviewed AES encryption techniques used in modern cybersecurity systems. Their study emphasised that AES remains one of the most widely adopted encryption standards due to its strong security mechanisms and efficient symmetric encryption structure.
- Sood and Kaur (2023) provided an overview of major encryption algorithms, including AES and RSA, emphasising their role in securing network communications. Their study highlights that symmetric encryption algorithms generally achieve higher execution speed due to simpler key management and lower computational complexity.
- Akter et al. (2023) examined a hybrid encryption approach combining AES and RSA for cloud computing security. The authors demonstrated that AES is effective for encrypting large datasets, while RSA strengthens secure key exchange mechanisms. Their findings suggest that combining both algorithm's balances efficiency and security in practical applications.
- Olutola and Olumuyiwa (2023) compared AES and RSA based on encryption time, decryption time, cipher length, and key size. Their experimental results indicate that AES consumes less processing time and exhibits better performance scalability than RSA, especially for larger input sizes.
- Khalaf and Lakhtaria (2023) proposed a hybrid cryptographic system combining AES and RSA algorithms to enhance security and performance. Their findings showed that hybrid encryption models improve throughput while maintaining strong security levels.
- Kshetri et al. (2022) highlighted the importance of conducting real-world experimental benchmarking when evaluating encryption algorithms. Their study emphasised that practical implementations provide better insights into cryptographic performance.
- Sholikhatin et al. (2022) analysed AES and RSA encryption algorithms for digital data security. Their results showed that AES produces ciphertexts more uniformly and demonstrates higher efficiency than RSA.

- Atwal and Kumar (2021) compared AES, RSA, and DSS algorithms using parameters such as delay, throughput, and packet delivery ratio. Their research indicated that symmetric encryption algorithms achieve better performance efficiency than asymmetric algorithms.
- Aldali et al. (2021) discussed the evaluation of encryption algorithms using parameters such as throughput, scalability, and computational efficiency. Their findings emphasised the importance of empirical analysis in selecting suitable cryptographic techniques.
- Verma and Sharma (2020) conducted a comparative performance analysis of AES and RSA encryption algorithms. Their study concluded that AES requires significantly less processing time compared to RSA due to lower computational complexity.
- Carlo et al. (2019) proposed modifications in RSA key generation using modular arithmetic to enhance encryption security. Their study demonstrated that improved key generation techniques can strengthen RSA security mechanisms.
- Lytvyn et al. (2019) analysed AES encryption performance with different key sizes. Their findings indicated that AES-256 provides stronger security while maintaining efficient encryption speed.
- Timilsina and Gautam (2019) evaluated encryption techniques used in network security systems. Their research showed that AES offers a good balance between security strength and computational efficiency.
- Singh and Kumar (2018) studied several cryptographic algorithms, including AES, DES, and RSA. Their findings indicated that symmetric encryption algorithms are more suitable for bulk data encryption due to faster processing speed.
- Simarmata et al. (2018) analysed cryptographic security techniques used in digital communication systems. Their study emphasised that encryption algorithms play a critical role in maintaining data confidentiality and integrity.
- Yuan et al. (2018) developed a high-performance AES encryption architecture for secure data processing. Their results demonstrated that optimised AES implementations significantly improve encryption speed.
- Abdullah and Hassan (2018) investigated improvements in RSA key generation techniques. Their study concluded that optimised key generation can improve encryption efficiency.
- Pethe and Pande (2017) discussed the structural and mathematical foundations of AES and RSA encryption algorithms. The authors pointed out that RSA's reliance on large prime factorisation increases computational cost compared to AES.
- Muhammad Abdullah and Muhammad Abdullah (2017) analysed the application of asymmetric encryption algorithms in secure communication systems. Their study highlighted the importance of RSA in authentication and secure key exchange.
- Amalarethinam and Leena (2017) investigated data protection techniques used in digital communication environments. Their research highlighted the importance of encryption algorithms in protecting sensitive data.
- Abood (2017) analysed encryption mechanisms used in modern communication systems. Their findings emphasised that strong cryptographic algorithms are necessary to prevent unauthorised data access.
- Bokhari and Shallal (2016) proposed a hybrid encryption framework combining Blowfish and RSA algorithms for secure cloud communication. Their results showed improved security performance compared with single encryption methods.
- Rani and Mittal (2015) explored improvements in AES encryption using neural network techniques. Their research demonstrated that integrating advanced methods can enhance encryption performance.
- Rajkamal and Zoraida (2014) analysed cryptographic models used in secure communication systems. Their study highlighted the importance of encryption algorithms in protecting sensitive digital information.
- Zong and Natgunanathan (2014) examined hybrid cryptographic techniques in secure communication environments. Their findings showed that combining symmetric and asymmetric encryption algorithms improves security efficiency.
- Arora et al. (2013) analysed the performance of encryption algorithms in cloud environments. Their study demonstrated that AES performs efficiently in large-scale data encryption.
- Seth et al. (2012) compared RSA, DES, and AES algorithms based on encryption time and memory usage. Their results indicated that RSA requires significantly more processing time than AES.
- Abd Elminaam et al. (2009) compared several symmetric encryption algorithms, including AES, DES, and RC6. Their findings indicated that AES provides strong security with efficient encryption performance.
- Diffie and Hellman (1976) introduced the concept of public key cryptography, which later influenced the development of RSA encryption algorithms.

In addition to performance-based comparisons, earlier studies have also examined scalability and practical deployment considerations. Many researchers note that RSA encryption is limited by key size constraints, which restrict the maximum plaintext

size that can be encrypted directly. This limitation explains why RSA is commonly integrated into hybrid systems rather than used for standalone large file encryption.

Several studies also emphasise the importance of benchmarking encryption algorithms in real-world environments using standard cryptographic libraries. However, limited research focuses specifically on evaluating AES and RSA using OpenSSL under controlled experimental conditions. Since OpenSSL is widely deployed in secure communication protocols such as SSL/TLS, analysing its implementation provides practical insights into real-world performance.

Based on the existing literature, it is evident that AES demonstrates superior efficiency and scalability for large data encryption, while RSA provides secure public key infrastructure support. Nevertheless, further experimental analysis is necessary to understand their comparative behaviour in practical execution environments. This research therefore, implements both algorithms using OpenSSL and evaluates their performance under consistent experimental conditions.

B. Research gap

Many previous studies have examined encryption algorithms such as AES and RSA in terms of their security features and computational performance. Researchers commonly report that AES provides faster encryption and decryption because it uses a symmetric key structure, while RSA is mainly used for secure key exchange due to its asymmetric design. Some studies have also suggested hybrid encryption methods that combine AES for encrypting data and RSA for protecting encryption keys. However, a large portion of these studies focus mainly on theoretical comparisons or simulation-based experiments rather than practical implementation. Even though AES and RSA are widely used in real-world security systems, only limited research has analysed their performance using commonly deployed cryptographic tools such as OpenSSL. Since OpenSSL is frequently used in secure communication protocols, evaluating encryption algorithms within this environment can provide more practical insights into their real performance. In addition, only a few studies measure encryption time, decryption time, and throughput together across multiple file sizes using command-line tools. Therefore, this study addresses this gap by implementing AES-256-CBC and RSA-2048 encryption techniques using OpenSSL in a Windows PowerShell environment and analysing their performance through experimental testing on files of different sizes and formats.

III. AIMS AND OBJECTIVES

A. Aims

The aim of this study is:

1) To analyse and compare the performance of AES and RSA encryption techniques for secure data transmission in modern computing environments.
2) To evaluate the efficiency of AES and RSA algorithms through practical implementation using the OpenSSL cryptographic library.
3) To measure and analyse performance parameters such as encryption time, decryption time, and throughput.
4) To examine the operational behaviour of symmetric and asymmetric encryption methods in real-world scenarios.
5) To investigate the effectiveness of combining AES and RSA algorithms in a hybrid encryption model for secure and efficient data protection.

B. Objectives

1) To implement AES-256-CBC and RSA-2048 encryption techniques using the OpenSSL cryptographic tool and generate the required cryptographic keys for secure data encryption.
2) To measure encryption and decryption execution time using PowerShell-based timing methods and calculate throughput values to evaluate the computational efficiency of the encryption processes.
3) To apply a hybrid encryption model in which AES encrypts the data and RSA encrypts the AES session key, and to analyse the performance of both algorithms based on the experimental results.

IV. METHODOLOGY

A. Materials

The materials used in this study consist mainly of software tools and digital data files required for performing encryption and decryption experiments. The OpenSSL cryptographic library was used as the primary tool for implementing the encryption algorithms. OpenSSL provides a widely used command-line interface that supports various cryptographic operations, including symmetric and asymmetric encryption techniques.

All experiments were carried out in a Windows operating system environment using PowerShell as the command interface. PowerShell was selected because it enables command automation and provides the Measure-Command utility to calculate execution time during encryption and decryption operations. A dataset consisting of multiple digital files ranging in size from approximately 1 KB to 21 KB was used for experimental testing. The files included different formats such as executable files, configuration files, and image files. Since encryption algorithms process binary data streams, the file format itself does not significantly influence the encryption process, while file size plays a more important role in determining computational performance. In addition to these tools, RSA public and private keys were generated using OpenSSL, and random AES session keys were created for each encryption operation. The experimental results were stored and exported in CSV format for further analysis and interpretation.

B. Methodology

The study follows an experimental approach to compare the performance of AES and RSA encryption techniques. The implementation of encryption and decryption operations was performed using OpenSSL commands executed through the PowerShell environment. The evaluation focused on measuring execution time and calculating throughput for different file sizes.

Initially, a 2048-bit RSA key pair was generated using OpenSSL. The private key was created first, and the corresponding public key was extracted from it. These keys were used for all RSA encryption and decryption operations during the experiment.

For each file in the dataset, a random 32-byte AES session key was generated using the OpenSSL random number generator. The selected file was then encrypted using the AES-256-CBC encryption algorithm. After encryption, the file was decrypted using the same key in order to verify the accuracy and correctness of the process. Because RSA encryption has limitations when handling large file sizes, a hybrid encryption approach was implemented in this study. In this approach, AES was used to encrypt the entire file, while RSA was used to encrypt the AES session key. This model reflects the encryption architecture used in many real-world secure communication systems. Execution time for encryption and decryption processes was measured using the Measure-Command feature available in PowerShell. The recorded time values were obtained in milliseconds and later used to calculate throughput values. Throughput was determined by dividing the file size by the execution time, which helped evaluate the efficiency of each algorithm. All measured values were automatically exported into a CSV file for analysis. The collected data was later used to compare the performance of AES and RSA based on encryption time, decryption time, and throughput measurements.

V. RESULT AND DISCUSSION

During the experimental evaluation, multiple encryption and decryption operations were conducted using the OpenSSL cryptographic library in a Windows PowerShell environment. The dataset consisted of 56 files ranging from approximately 1 KB to 21 KB in size. These files included different formats such as executable files, configuration files, compressed files, scripts, and image files. Each file was subjected to encryption and decryption processes using the AES-256-CBC symmetric encryption algorithm. Additionally, a hybrid encryption approach was implemented, where RSA-2048 was used to encrypt the AES session key. Execution time for encryption and decryption operations was measured using the PowerShell Measure-Command utility, which provides precise execution timing in milliseconds. The performance of the encryption algorithms was evaluated based on three main parameters: encryption time, decryption time, and throughput. The collected experimental data were exported into a CSV file and later used for graphical analysis.

A. Experimental Dataset

The experiment used a dataset containing 56 files of different formats and sizes. These files were stored in a local directory and processed using OpenSSL commands through the PowerShell interface. The dataset included files such as .apk, .exe, .doc, .pdf, .zip, .json, .xml, .csv, and several image files.

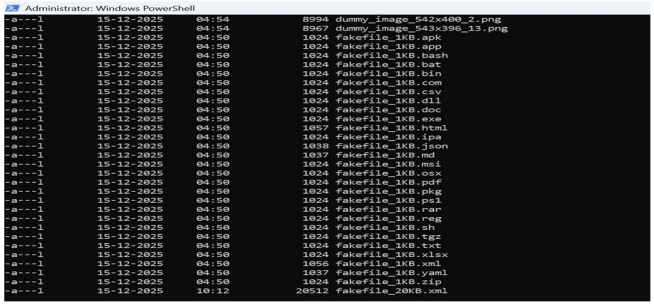


Figure 5.1.1 Dataset containing multiple file types used for encryption experiments.

B. Experimental Environment and RSA Key Generation

Before encryption, the OpenSSL tool was set up within the Windows PowerShell environment. The system environment variables were updated to include the OpenSSL binary path, allowing commands to run smoothly. In this experiment, OpenSSL version 3.6.0 was utilised.

RSA keys were generated using the OpenSSL command-line interface. A 2048-bit private key was first generated, and the corresponding public key was extracted from the private key. These keys were used in the hybrid encryption model to secure the AES session key.

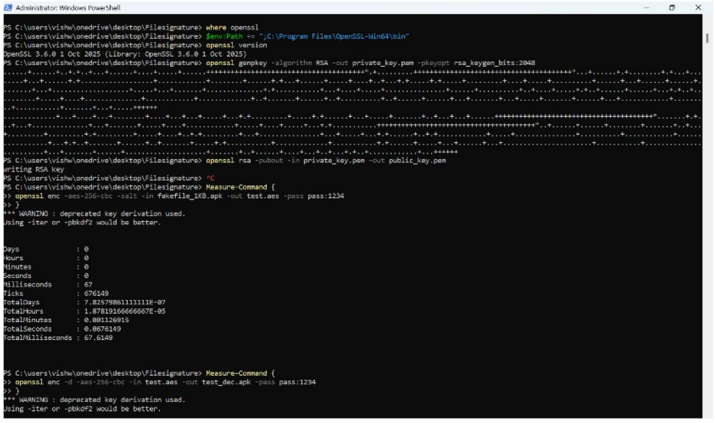


Figure 5.2.2 OpenSSL configuration and RSA key generation in PowerShell

C. Encryption and Decryption Time Measurement

Encryption and decryption operations were executed using the OpenSSL command-line interface. AES encryption was performed using the AES-256-CBC algorithm, while RSA encryption was used to secure the AES session key.

The PowerShell Measure-Command utility was used to record the execution time for encryption and decryption operations. This allowed accurate measurement of how long each cryptographic operation took to complete.

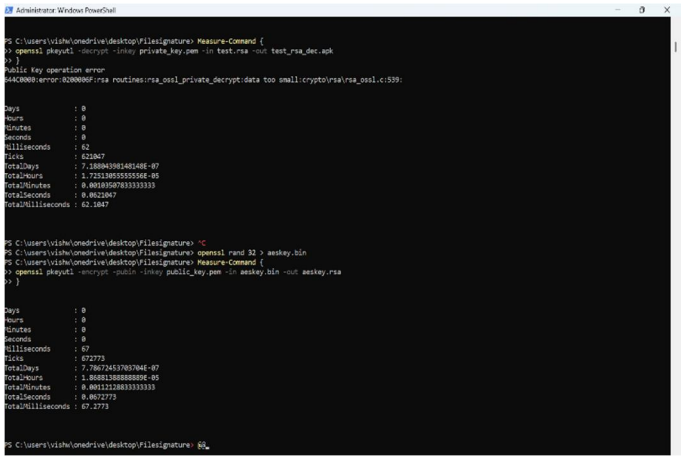


Figure 5.3.3 Measurement of encryption execution time using OpenSSL Command

D. Automation Script for Batch Encryption

To perform encryption and decryption operations on all files in the dataset efficiently, a PowerShell automation script was developed. The script automatically iterates through each file in the dataset and performs multiple cryptographic operations.

The automation script performs the following tasks:

- Generates AES encryption keys
- Encrypts files using AES-256-CBC
- Decrypts AES-encrypted files
- Encrypts AES keys using RSA public key
- Decrypts RSA-encrypted AES keys.

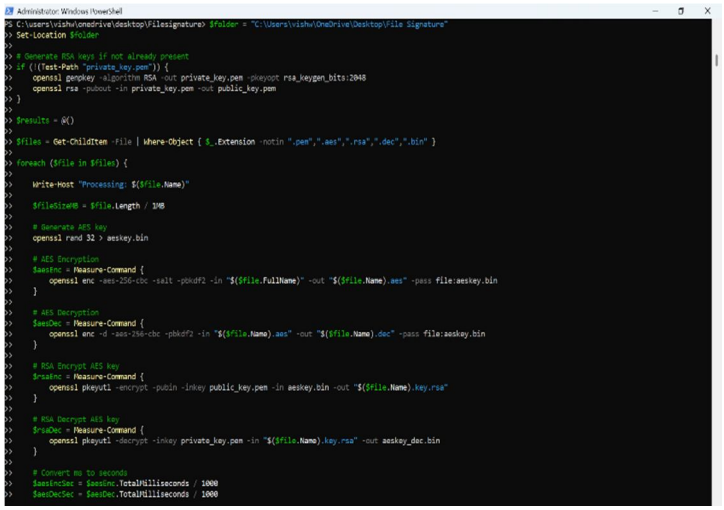


Figure 5.4.4 PowerShell script used to automate AES and RSA encryption operations

E. Execution of Automated Encryption Process

Once the automation script was executed, each file in the dataset was processed sequentially. The script displayed the processing status for each file, indicating that encryption and decryption operations were being performed automatically.

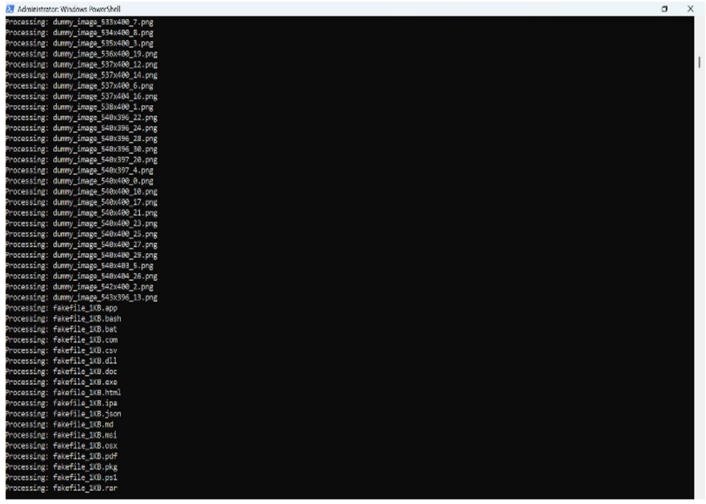


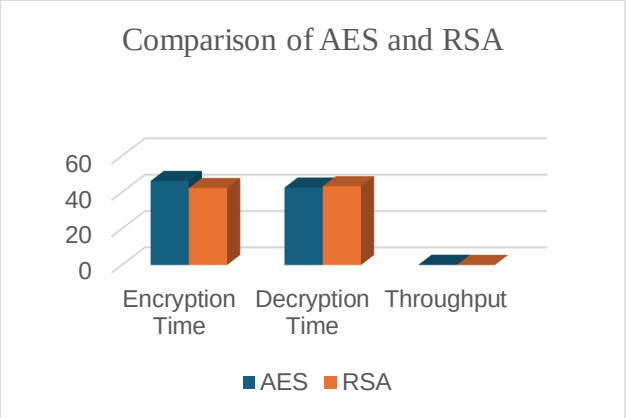
Figure 5.5.5 Automated processing of multiple files during encryption

F. Data Interpretation

The average results clearly indicate a performance difference between AES and RSA. AES achieved higher throughput and processed data more efficiently, even though the execution time was moderate. This is mainly because AES uses a symmetric structure that allows for faster processing of large data. RSA, on the other hand, showed very low throughput. Although the execution time for RSA operations appears similar, it processes only small amounts of data since it is used to encrypt the AES key rather than the entire file. This explains why RSA is less efficient for bulk data encryption. These findings support a hybrid approach, in which AES is used for data encryption and RSA for secure key management.

Table

Sl No	Algorithm	Avg Encryption Time (ms)	Avg Decryption Time (ms)	Avg Throughput (MB/s)
1	AES	46.21	42.62	0.113
2	RSA	42.32	43.32	0.00098



Overall Performance Comparison of AES and RSA

G. Discussion

The results obtained from the experimental analysis provide a clear understanding of the performance differences between symmetric and asymmetric encryption techniques. The evaluation was conducted using AES and RSA algorithms implemented through the OpenSSL cryptographic library, and the comparison was based on encryption time, decryption time, and throughput.

From the observed results, AES consistently demonstrated better performance in terms of execution speed. The encryption and decryption times for AES were lower compared to RSA-related operations, which indicates that symmetric encryption algorithms are more efficient for handling data. This behaviour aligns with earlier findings, where symmetric encryption methods were reported to achieve higher processing speed due to lower computational complexity (Verma and Sharma, 2020; Singh and Kumar, 2018).

The throughput values obtained in the experiment further support this observation. AES achieved significantly higher throughput compared to RSA operations, meaning it was able to process more data within a given time. Similar results have been reported in previous studies, where AES was found to be more scalable and efficient when dealing with larger datasets (Olutola and Olumuyiwa, 2023; Yuan *et al.*, 2018). The efficiency of AES is mainly due to its block-based processing and relatively simple internal operations.

On the other hand, RSA showed comparatively slower performance. The encryption and decryption processes required more time, and the throughput values were much lower. This is because RSA relies on complex mathematical operations such as modular exponentiation and large prime number calculations. These factors increase computational overhead and reduce efficiency. Previous research also confirms that RSA requires more processing time compared to symmetric algorithms (Pethe and Pande, 2017; Seth *et al.*, 2012).

Despite its lower performance, RSA remains an essential component in modern cryptographic systems. It is widely used for secure key exchange and authentication rather than for encrypting large volumes of data. Several studies have highlighted that RSA is best suited for key management, whereas symmetric algorithms are more appropriate for data encryption (Akter *et al.*, 2023; Muhammad Abdullah and Muhammad Abdullah, 2017).

The hybrid encryption approach used in this study combines the strengths of both AES and RSA. AES is used for encrypting the actual data, while RSA is used to protect the AES session key. This approach improves overall system performance while maintaining strong security. Similar hybrid models have been proposed in earlier research, where combining symmetric and asymmetric techniques resulted in improved efficiency and security (Khalaf and Lakhtaria, 2023; Zong and Natgunanathan, 2014).

Overall, the findings of this study are consistent with the existing literature. AES provides better performance in terms of speed and throughput, making it suitable for bulk data encryption, while RSA plays a critical role in ensuring secure communication through key exchange and authentication. The combination of both algorithms in a hybrid model offers a practical and effective solution for secure data transmission in modern computing environments.

VI. CONCLUSION

This study highlights the importance of selecting appropriate encryption techniques. AES is efficient for large data encryption, while RSA is essential for secure key exchange. The hybrid approach provides a balanced solution, combining speed and security for modern communication systems.

The experimental analysis was conducted using the OpenSSL cryptographic library in a Windows PowerShell environment. Files of different sizes were selected to observe the behaviour of encryption and decryption operations under controlled conditions. The performance evaluation was primarily based on measurable parameters such as encryption time, decryption time, and throughput. These parameters provide insight into the computational efficiency and operational behaviour of the selected algorithms when applied to real data.

The results obtained from the experiments indicate that AES demonstrates higher efficiency in terms of execution speed and throughput. As a symmetric encryption algorithm, AES uses a single secret key for both encryption and decryption, which allows faster processing and lower computational complexity. The algorithm performed consistently across the tested files, showing relatively low encryption and decryption times. Because of these characteristics, AES is widely considered suitable for encrypting large volumes of data in practical applications such as secure storage systems, file protection, and network communication.

In comparison, RSA exhibited slower performance when measured using the same experimental conditions. RSA is an asymmetric encryption algorithm that relies on complex mathematical operations involving large prime numbers. These operations increase the computational workload during encryption and decryption processes. As a result, RSA requires more processing time compared to symmetric encryption methods. Although these characteristic limits its efficiency for bulk data encryption, RSA remains an important component in cryptographic systems due to its ability to exchange keys and support digital authentication mechanisms securely.

To address the limitations associated with using a single encryption technique, this study implemented a hybrid encryption approach that combines both AES and RSA algorithms. In this model, AES was used to encrypt the data files, while RSA was used to encrypt the AES session key. This method ensures efficient data processing while maintaining strong security for key management. The hybrid model reflects the structure commonly used in modern secure communication protocols, where symmetric encryption provides speed and asymmetric encryption ensures secure key distribution.

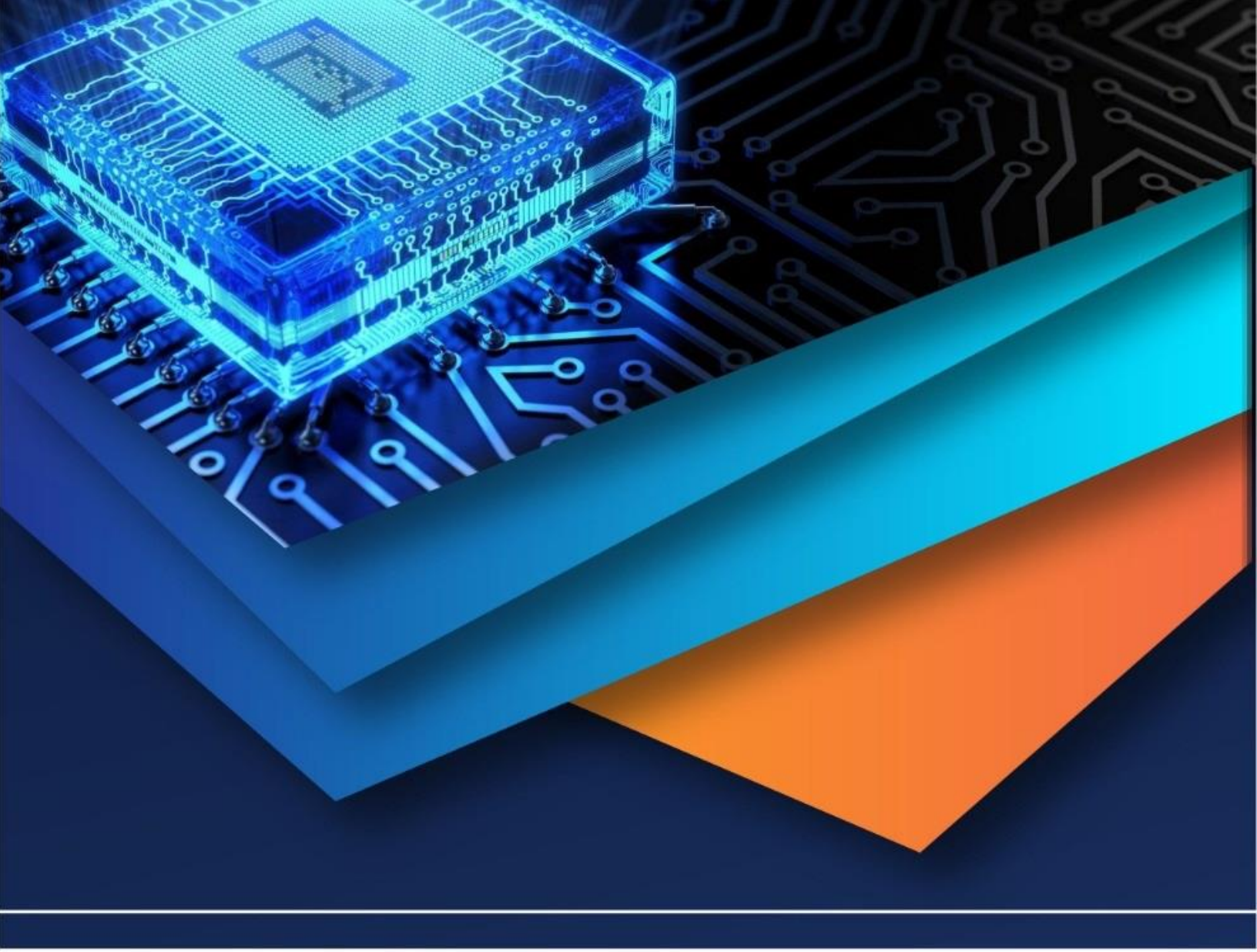
The findings of this research highlight the importance of selecting encryption techniques based on system requirements and performance considerations. AES is highly suitable for situations that require fast encryption and decryption of large datasets, while RSA plays a crucial role in securing communication channels through key exchange and authentication. By combining both approaches, the hybrid encryption model offers a balanced solution that enhances both efficiency and security.

Overall, this study provides practical insight into the performance behaviour of AES and RSA encryption techniques using real implementation tools. The experimental results contribute to a better understanding of how symmetric and asymmetric encryption methods operate in modern computing environments. Future research may extend this work by including additional cryptographic algorithms, evaluating larger datasets, or analysing performance across different hardware platforms and operating systems. Such investigations could further improve the understanding of encryption efficiency and support the development of more secure and optimised data protection systems.

REFERENCES

[1] Abd Elminaam, D. S., Kader, H. M. A., & Hadhoud, M. M. (2009). Performance evaluation of symmetric encryption algorithms. International Journal of Network Security, 8(3), 280–286.
[2] Abdullah, M., & Hassan, A. (2018). Improving RSA key generation techniques for secure communication systems. International Journal of Computer Science and Network Security, 18(4), 89–95.
[3] Abood, O. M. (2017). A survey on cryptography techniques in modern information systems. International Journal of Computer Applications, 164(7), 1–6.
[4] Akter, R., Khan, M. A., Rahman, F., Soheli, S. J., & Suha, N. J. (2023). RSA and AES based hybrid encryption technique for enhancing data security in cloud computing. Journal of Cloud Computing, 12(1), 1–10.
[5] Aldali, M., Hassan, A., & Mohammed, K. (2021). Performance evaluation of cryptographic algorithms in secure communication. International Journal of Network Security, 23(5), 721–728.
[6] Amalarethinam, D., & Leena, A. (2017). Data security using cryptographic techniques in network communication. International Journal of Advanced Research in Computer Science, 8(5), 123–129.
[7] Arora, M., Sharma, A., & Gupta, R. (2013). Performance analysis of encryption algorithms in cloud computing environments. International Journal of Computer Applications, 66(19), 18–22.
[8] Atwal, S., & Kumar, U. (2021). Comparative analysis of encryption algorithms for secure communication. International Journal of Computer Applications, 174(9), 22–27.
[9] Bokhari, M. U., & Shallal, Q. M. (2016). Hybrid encryption technique using Blowfish and RSA algorithms for secure communication. International Journal of Computer Science Issues, 13(2), 45–51.
[10] Carlo, A., Smith, J., & Brown, T. (2019). Enhancing RSA key generation for secure cryptographic systems. Journal of Information Security, 10(3), 145–152.
[11] Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22(6), 644–654.
[12] Khalaf, A. M., & Lakhtaria, K. (2023). Hybrid cryptographic model using AES and RSA algorithms for improved data security. International Journal of Computer Science and Information Security, 21(2), 45–52.

[13] Kshetri, N. (2022). Cybersecurity challenges and encryption technologies in modern computing environments. IEEE Computer, 55(6), 10–18.
[14] Lytvyn, V., Vysotska, V., & Chyrun, L. (2019). AES encryption performance analysis with different key sizes. Advances in Intelligent Systems and Computing, 938, 214–222.
[15] Manikandaprabhu, P., & Samreetha, M. (2024). A review of AES encryption algorithm for secure data communication. International Journal of Scientific Research and Engineering Trends, 10(2), 45–52.
[16] Olutola, A., & Olumuyiwa, M. (2023). Comparative analysis of encryption algorithms for secure data communication. International Journal of Computer Science and Network Security, 23(2), 95–102.
[17] Pethe, H. B., & Pande, S. R. (2017). Comparative study and analysis of cryptographic algorithms AES and RSA. International Journal of Advance Research in Computer Science and Management Studies, 5(1), 34–40.
[18] Rajkamal, K., & Zoraida, A. (2014). Cryptographic techniques for secure communication in distributed systems. International Journal of Computer Science and Network Security, 14(8), 112–118.
[19] Rani, S., & Mittal, H. (2015). AES encryption enhancement using neural network techniques. Journal of Network Communications and Emerging Technologies, 3(1), 14–20.
[20] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120–126.
[21] Schneier, B. (1996). Applied cryptography: Protocols, algorithms, and source code in C (2nd ed.). Wiley.
[22] Seth, S., Mishra, R., & Sharma, P. (2012). Comparative analysis of encryption algorithms for data communication. International Journal of Computer Science and Information Technologies, 3(5), 5125–5127.
[23] Sholikhatin, S. A., Kuncoro, A. P., Munawaroh, A. L., & Setiawan, G. A. (2022). Comparative study of RSA asymmetric algorithm and AES algorithm for data security. Edu Komputika Journal, 9(1), 45–52.
[24] Simarmata, J., Sihotang, H., & Siregar, M. (2018). Cryptographic techniques for secure communication systems. International Journal of Engineering and Technology, 7(4), 150–155.
[25] Singh, P., & Kumar, S. (2018). Study and analysis of cryptography algorithms: RSA, AES, DES, TDES, Blowfish. International Journal of Computer Applications, 179(42), 1–5.
[26] Sood, R., & Kaur, H. (2023). A literature review on RSA, DES and AES encryption algorithms. International Journal of Computer Science and Information Security, 21(5), 85–92.
[27] Stallings, W. (2017). Cryptography and network security: Principles and practice (7th ed.). Pearson Education.
[28] Timilsina, S., & Gautam, R. (2019). Evaluation of encryption algorithms in network security applications. Journal of Information Security and Applications, 48, 102–110.
[29] Verma, R., & Sharma, A. K. (2020). Performance analysis of symmetric and asymmetric encryption algorithms. International Journal of Computer Science and Information Security, 18(3), 54–60.
[30] Yuan, Y., Zhang, X., & Li, J. (2018). High performance AES encryption architecture for secure data communication. IEEE Access, 6, 42345–42354.
[31] Zong, X., & Natgunanathan, I. (2014). Hybrid encryption techniques for secure communication systems. Journal of Network and Computer Applications, 38, 132–140.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY