Basic and Applied Computing for Digital Innovation (BACODING)
Volume 01, No. 02, August, 2026, pp. 89-97
Received: 11 August 2026; Revised: 25 August 2026; Accepted: 27 August 2026
https://bacoding.id/index.php/journal/
DOI:

Research Article

A Comparative Analysis of AES and RSA Algorithms for Digital Data Security

Pandu Sujaba[1], Muhammad Fernanda[2], RG Guntur Alam[3]
[1,2,3,] Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Bengkulu

***Correspondence:** Psujaba@gmail.com

Abstract: The expansion of digital services has increased both the volume and the value of information that must be protected. Unauthorized disclosure, interception, alteration, and misuse of stored or transmitted data make confidentiality an important security requirement. Cryptography addresses this requirement by converting readable information into an encoded form and recovering it through a corresponding decryption procedure. This study examines the Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) by comparing their structures, key mechanisms, computational characteristics, and practical roles in digital security. A descriptive comparative method was applied by examining relevant literature and the performance measurements used in this study. AES represents symmetric encryption because the same secret key is involved in both directions of the process, whereas RSA employs a public-key and private-key pair. The measured results show that AES-128 completed encryption in 5.8 ms/character and decryption in 27.2 ms/character, while RSA required 8.7 ms/character and 47.6 ms/character, respectively. These findings indicate that AES is the more efficient option for bulk data encryption, while RSA is better suited to asymmetric tasks such as key establishment and digital signatures. A combined scheme can therefore use AES for the data payload and RSA for protecting the session key, reducing the principal key-distribution weakness of symmetric encryption.

Keywords: AES, RSA, cryptography, data confidentiality, encryption, digital security

1. Introduction

The rapid advancement of information technology has accelerated the volume of digital data exchange over the Internet, making data security a primary priority in modern information systems. Threats to data confidentiality, integrity, and authenticity—such as eavesdropping, tampering, and unauthorized interception—necessitate the implementation of robust protection mechanisms. Cryptography addresses these challenges by securing transmitted information against unauthorized access and alteration. On a global scale, this requirement becomes increasingly critical as the volume of data stored in interconnected digital environments grows, demanding secure processing and storage frameworks [1]-[3]

Cryptographic algorithms are generally categorized into two main paradigms: symmetric and asymmetric cryptography. The Advanced Encryption Standard (AES) represents a widely adopted symmetric algorithm, offering superior processing speed. Conversely, Rivest–Shamir–Adleman (RSA) is an asymmetric algorithm that utilizes a pair of public and private keys, making it preferred for key

distribution and authentication. However, despite its high security for key exchange, asymmetric cryptography like RSA requires significantly higher computational resources and processing time compared to symmetric algorithms. This trade-off presents a dilemma for system developers, who must choose between the speed of AES (with its complex key distribution) and the security of RSA (with its heavy computational overhead). [4]-[5]

Prior national and international studies have extensively evaluated the comparative performance of these two algorithms across various domains. In text data processing, experimental findings demonstrate that AES achieves substantially faster encryption and decryption times than RSA, particularly for large datasets, whereas RSA provides robust security for key exchange despite its higher computational cost. Similar patterns emerge in digital image processing, where AES exhibits superior processing speed, while RSA encryption tends to yield lower image reconstruction quality, as indicated by higher Mean Squared Error (MSE) values [6]-[7]

International studies further reinforce these observations. In cloud computing contexts, AES is often recommended as the primary cryptographic method for data encryption and storage due to its lower resource and power consumption relative to RSA. Comparative evaluations confirm that AES outperforms RSA in both encryption and decryption speeds while leaving a smaller unencrypted data footprint. Comprehensive analyses of encryption techniques emphasize that factors such as encryption strength, computational complexity, key management, scalability, and flexibility must be holistically evaluated to determine the suitability of AES and RSA for secure communication.[8]-[11]

To mitigate individual algorithmic limitations, several researchers have proposed hybrid AES–RSA schemes. These approaches utilize RSA to encrypt the secret key while employing AES to encrypt the primary data payload, thereby subjecting transmitted messages to a two-tier encryption and decryption process to enhance security. Literature reviews support this hybrid approach, concluding that AES provides superior encryption speed and efficiency, whereas RSA ensures robust key security at the expense of higher memory overhead. Consequently, a hybrid AES–RSA cryptosystem leverages the computational efficiency of AES alongside the key-management strengths of RSA [12] – [14]

Nevertheless, most existing studies focus primarily on a single data type (either text or image) using limited evaluation parameters, leaving algorithmic overhead on micro-sized datasets relatively underexplored. A literature review of national journals published between 2016 and 2025 confirms that while symmetric algorithms like AES excel in efficiency and speed, they remain vulnerable in key distribution. Conversely, asymmetric algorithms like RSA offer strong key exchange security but demand greater computational resources and execution time, justifying the recommendation for hybrid cryptosystems to balance efficiency and security [15]-[16]

Based on these considerations, further research is required to comprehensively analyze and compare the performance of AES and RSA across multiple parameters—including encryption/decryption execution times, computational complexity, resource consumption, and theoretical security levels. Such an analysis aims to provide objective recommendations for selecting the most suitable cryptographic algorithm tailored to specific digital security scenarios across text, image, and large document datasets.

2. Research Method

2.1 AES Algorithm Analysis

AES uses a shared secret key for the forward and reverse transformations. Its basic data unit is a 128-bit block, while the supported key sizes are 128, 192, and 256 bits. Encryption applies a sequence of byte substitution, row shifting, column mixing, and round-key addition; decryption uses the corresponding inverse transformations [2].

Table 1. Characteristics of the AES Algorithm

No.	Parameter	AES
1	Algorithm type	Symmetric
2	Block size	128 bit
3	Key length	128 bit, 192 bit, and 256 bit
4	Key	Secret key
5	Encryption process	SubBytes, ShiftRows, MixColumns, AddRoundKey
6	Decryption process	InvSubBytes, InvShiftRows, InvMixColumns, AddRoundKey
7	Speed	Relatively fast
8	Efficiency	Relatively high
9	Large-data security	Suitable
10	Limitation	Secret-key distribution

2.2 AES Encryption Process

During AES encryption, the input state is first combined with round-key material and then processed through repeated transformation rounds. SubBytes changes byte values through a substitution table, ShiftRows rearranges positions, and MixColumns combines information within each column. AddRoundKey incorporates the round-specific key. The final round excludes the column-mixing stage. Because both parties use the same secret, the key itself must be transferred and stored through a secure mechanism [2], [8].

2.3 AES Usage Analysis

AES is well suited to files, databases, and communication streams in which many data blocks have to be processed. Its main practical strength in this context is the relatively low computational overhead of symmetric processing. The principal operational concern is the secure distribution and protection of the shared key [2], [6].

2.4 RSA Algorithm Analysis

RSA separates the encryption and decryption roles by assigning them to a public key and a private key. The public component can be distributed, whereas the private component remains under the control of the key owner. This arrangement distinguishes RSA from the single-secret-key model used by AES [1], [2].

RSA key construction relies on number-theoretic operations involving large integers and prime factors. The generated key pair is mathematically related, while the security assumption is associated with the difficulty of recovering the private information from the public parameters under appropriate key sizes [8].

Table 2. Characteristics of the RSA Algorithm

No.	Parameter	RSA
1	Algorithm type	Asymmetric
2	Key type	Public key and private key
3	Encryption key	Public key
4	Decryption key	Private key
5	Operation basis	Modular arithmetic
6	Key generation	Uses prime numbers
7	Speed	Relatively slower than AES
8	Large data	Less efficient
9	Key distribution	More practical using a public key
10	Application	Key exchange, limited encryption, a

2.5 RSA Key Generation

RSA key generation begins with the selection of two distinct prime numbers, p and q. Their product forms the modulus used by the public and private keys:

$$n = p \times q \quad (1)$$

Here, p and q denote the selected prime numbers and n represents the resulting modulus. The modulus is then used to obtain Euler's totient value:

$$\varphi(n) = (p - 1)(q - 1) \quad (2)$$

A public exponent e is selected so that it is relatively prime to the totient. The private exponent d is calculated as the modular inverse of e under the same modulus relationship:

$$d \times e \equiv 1 \pmod{\varphi(n)} \quad (3)$$

The public key is represented by (e, n), while the private key is represented by (d, n). The public component can be distributed for the intended cryptographic operation; the private component must remain confidential because possession of it enables the corresponding private-key operation [8].

2.6 Comparison Parameters

For the comparison, AES and RSA were examined using properties that directly influence their practical use: algorithm class, key arrangement, processing time, computational efficiency, suitability for large data, and key-management requirements.

Table 3. Comparison Parameters of AES and RSA

No.	Parameter	AES	RSA
1	Algorithm type	Symmetric	Asymmetric
2	Key mechanism	Single secret key	Public key and private key
3	Key length	128, 192, and 256 bit	Generally uses much larger key sizes
4	Block size	128 bit	Does not use a block structure like AES
5	Processing speed	Relatively fast	Relatively slower
6	Resource efficiency	Relatively high	Relatively greater resource requirement
7	Large data	Highly suitable	Less suitable for direct encryption
8	Key distribution	Requires additional protection	More suitable because of public key
9	Main data encryption	Highly suitable	Less efficient
10	Key protection	Not the primary function	Highly suitable
11	Digital signatures	Not the primary function	Can be used
12	Application	Files, databases, and data communication	

2.7 Processing Speed Parameter

Processing time was selected as a direct indicator of computational efficiency. The measured encryption and decryption times provide a basis for comparing how much processing is required by AES-128 and RSA under the test conditions.

Table 4. AES and RSA Processing Time Comparison

Process	AES-128	RSA
Encryption	5.8 ms/character	8.7 ms/character
Decryption	27.2 ms/character	47.6 ms/character

2.8 Efficiency Parameter

AES has a relatively lightweight processing model because the same shared secret is used throughout the symmetric operation. RSA performs more computationally intensive public-key arithmetic, so its direct use for large payloads is less efficient. For this reason, RSA is generally more appropriate for supporting functions such as key establishment or protection than for bulk encryption [1], [6].

2.9 Key Management Parameter

The main key-management drawback of AES is the need to deliver the shared secret to every authorized participant. A secure distribution method is therefore required before encrypted information can be exchanged.

RSA addresses this distribution issue through its two-key arrangement. A public key can be published, while the private key remains secret. This separation makes RSA useful for establishing or protecting the symmetric key used by a bulk-encryption algorithm [2], [8].

2.10 Application Parameter

Considering the complete set of parameters, AES is the stronger choice for encrypting the actual data payload because of its processing efficiency. RSA is more appropriate for asymmetric operations, particularly key establishment and related security functions. A hybrid design can combine the two: AES protects the data, while RSA protects the AES key. Such an arrangement retains the speed advantage of symmetric encryption while reducing the key-distribution difficulty [1], [2].

3. Results and Discussion

3.1 Results of AES Algorithm Analysis

The AES assessment confirms that the algorithm uses one shared secret for both directions of the cryptographic process. Its 128-bit block structure and three supported key lengths provide a standardized framework for protecting digital data [2]. Under the characteristics examined in this study, AES offers a favorable processing profile for high-volume data. The same property that makes it efficient, however, also creates the requirement that the shared secret be delivered securely to every legitimate party [2], [8].

Table 5. AES Algorithm Analysis Results

No.	Parameter	Analysis Result
1	Algorithm type	Symmetric
2	Block size	128 bit
3	Key length	128 bit, 192 bit, and 256 bit
4	Key	Secret key
5	Encryption process	SubBytes, ShiftRows, MixColumns, AddRoundKey

6	Decryption process	InvSubBytes, InvShiftRows, InvMixColumns, AddRoundKey
7	Speed	Relatively fast
8	Efficiency	Relatively high
9	Large-data security	Suitable
10	Limitation	Secret-key distribution

The test data indicate that AES-128 completed the two measured operations in less time than RSA. The individual values are presented below.

Table 6. AES-128 Processing Time Results

Process	Time
Encryption	5.8 ms/character
Decryption	27.2 ms/character

The measured AES-128 times were 5.8 ms/character for encryption and 27.2 ms/character for decryption. The two values show that the decryption operation took longer than encryption in the observed test. These measurements provide the AES baseline for the corresponding RSA evaluation and allow the relative processing cost of the two algorithms to be quantified.

From the observed results, AES demonstrates a practical advantage when the task involves repeated protection of a large data payload. Its symmetric construction avoids the heavier public-key arithmetic associated with RSA [1], [6]. The trade-off is the shared-secret requirement. Both endpoints must obtain the same AES key, so key distribution and secure storage remain important implementation concerns [8]. A practical architecture can address this limitation by pairing AES with an asymmetric mechanism. In such a design, AES handles the payload and RSA is used only to protect or establish the AES key.

3.2 Results of RSA Algorithm Analysis

RSA was evaluated as the asymmetric counterpart to AES. Its two-key arrangement separates the publicly distributed component from the confidential private component, giving it a different operational role in a security system [1], [2]. RSA relies on modular arithmetic involving large integers. The public and private components perform complementary operations, which gives the algorithm its characteristic public-key behavior [1].

Table 7. RSA Algorithm Analysis Results

No.	Parameter	Analysis Result
1	Algorithm type	Asymmetric
2	Key type	Public key and private key
3	Encryption key	Public key
4	Decryption key	Private key
5	Operation basis	Modular arithmetic
6	Key generation	Uses prime numbers
7	Speed	Relatively slower than AES
8	Efficiency	Relatively lower than AES
9	Large data	Less suitable for direct encryption
10	Key distribution	More suitable
11	Key protection	Suitable
12	Application	Key exchange, authentication, and digital signatures

The public key can be shared with intended parties without exposing the private key. This makes RSA particularly useful when a system needs to establish or protect a secret without directly distributing

that secret in plaintext [2], [8]. The RSA measurements were higher than the corresponding AES-128 values. The observed encryption and decryption times are listed in the next table.

Table 8. RSA Processing Time Results

Process	Time
Encryption	8.7 ms/character
Decryption	47.6 ms/character

RSA required 8.7 ms/character for encryption and 47.6 ms/character for decryption in the observed test. Both figures exceed the corresponding AES-128 measurements. The higher processing cost limits the practicality of using RSA as the direct bulk-encryption mechanism for large data objects. Its strengths are better aligned with operations that specifically require asymmetric keys.

A major benefit of RSA is that the public key does not need to be kept secret. This permits a sender to use the recipient's public information without obtaining the recipient's private key. Beyond encryption, RSA can support digital-signature and authentication-related mechanisms. These roles complement rather than replace the bulk-data encryption role of AES. The trade-off is computational cost. The larger integer operations used by RSA increase the processing burden relative to the symmetric AES construction, especially when the algorithm is applied repeatedly to large data.

3.3 Performance Comparison of AES and RSA

The final comparison uses processing time because it provides a measurable indication of the computational cost of each algorithm under the same test scenario. For AES-128, the observed times were 5.8 ms/character for encryption and 27.2 ms/character for decryption. RSA recorded 8.7 ms/character and 47.6 ms/character for the same two operations.

Table 9. Performance Comparison of AES-128 and RSA

Process	AES-128	RSA	Time Difference
Encryption	5.8 ms/character	8.7 ms/character	2.9 ms/character
Decryption	27.2 ms/character	47.6 ms/character	20.4 ms/character

For encryption, the measured gap was 2.9 ms/character: AES-128 required 5.8 ms/character, whereas RSA required 8.7 ms/character. The decryption gap was larger at 20.4 ms/character. AES-128 recorded 27.2 ms/character compared with 47.6 ms/character for RSA.

Relative to the AES-128 measurements, the RSA value was about 50% higher for encryption and about 75% higher for decryption. Within this test, the result therefore favors AES in terms of processing efficiency.

Table 10. Percentage Difference in Processing Time

Process	AES-128	RSA	RSA Slower
Encryption	5.8 ms	8.7 ms	50%
Decryption	27.2 ms	47.6 ms	75%

The measurements consistently place AES-128 below RSA in processing time for both operations. This supports the use of AES when the main requirement is efficient protection of substantial data volumes. RSA's higher processing cost does not make it inferior in every security function. The algorithms address different architectural needs: AES is optimized for efficient symmetric data protection, whereas RSA provides the public-key mechanism required for functions such as key establishment and digital signatures.

4. 4. Conclusion

The comparison shows that AES-128 achieved the lower processing times in the observed test, with 5.8 ms/character for encryption and 27.2 ms/character for decryption. RSA required 8.7 ms/character and 47.6 ms/character for the same operations. These results support the selection of AES for bulk data encryption, while RSA is better positioned for asymmetric key-management tasks. Rather than treating the algorithms as substitutes, a hybrid arrangement can assign each algorithm to the function for which it is technically better suited: AES protects the data payload and RSA protects or establishes the AES key.

5. REFERENCES

[1]. Saputra, R. S., et al., (2026). "Analisis Komparasi Performansi Algoritma Kriptografi Rsa Dan Aes Pada Keamanan Data Teks". *JATI (Jurnal Mahasiswa Teknik Informatika)*. Vol 10 (2), pp. 1866-1870. Doi: https://doi.org/10.36040/jati.v10i2.17185

[2]. Putri, dkk. (2018) Analisis Kriptografi Simetris AES dan Kriptografi Asimetris RSA pada Enkripsi Citra Digital. *ETHOS: Jurnal Penelitian dan Pengabdian kepada Masyarakat, Universitas Islam Bandung*. Vol 6 (2), pp. 197-207 DOI: http://dx.doi.org/10.29313/ethos.v6i2.2909

[3]. Laurentinus, L., Pradana, H. A., Sylfania, D. Y., & Juniawan, F. P., "Performance Comparison of RSA and AES to SMS Messages Compression Using Huffman Algorithm," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 171–177, 2020, doi: 10.14710/jtsiskom.2020.13468.

[4]. Sholikhatin, S. A., Kuncoro, A. P., Munawaroh, A. L., & Setiawan, G. A., "Comparative Study of RSA Asymmetric Algorithm and AES Algorithm for Data Security," *Edu Komputika Journal*, vol. 9, no. 1, pp. 60–67, 2022, doi: 10.15294/edukomputika.v9i1.57389.

[5]. Fatima, S., Rehman, T., Fatima, M., Khan, S., & Ali, M. A., "Comparative Analysis of AES and RSA Algorithms for Data Security in Cloud Computing," *Engineering Proceedings*, vol. 20, no. 1, Art. 14, 2022, doi: 10.3390/engproc2022020014.

[6]. Xu, Y., Wu, S., Wang, M., & Zou, Y., "Design and Implementation of Distributed RSA Algorithm Based on Hadoop," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1047–1053, 2020.

[7]. Pattanavichai, S., "Program for Simulation and Testing of Apply Cryptography of Advanced Encryption Standard (AES) Algorithm with Rivest-Shamir-Adleman (RSA) Algorithm for Good Performance," *International Journal of Electronics and Telecommunications*, vol. 68, no. 3, pp. 475–481, 2022.

[8]. Commey, D., Klogo, S. G., & Gadze, J. D., "Performance Comparison of 3DES, AES, Blowfish and RSA for Dataset Classification and Encryption in Cloud Data Storage," *International Journal of Computer Applications*, vol. 177, no. 40, pp. 17–22, 2020.

[9]. Liu, J.-J., Tsang, K.-T., & Deng, Y.-H., "A Variant RSA Acceleration with Parallelization," *arXiv*, 2021.

[10]. Barker, E., "Recommendation for Key Management: Part 1 – General," *NIST Special Publication 800-57 Part 1 Revision 5*, National Institute of Standards and Technology, 2020.

[11]. Paar, C., & Pelzl, J., "Understanding Cryptography: A Textbook for Students and Practitioners," *Springer*, 2020.

[12]. Stallings, W., "Cryptography and Network Security: Principles and Practice," 8th ed., *Pearson*, 2020.

[13]. Katz, J., & Lindell, Y., "Introduction to Modern Cryptography," 3rd ed., *CRC Press*, 2020.

[14]. Singh, G., & Singh, A., "A Comparative Study of Symmetric and Asymmetric Cryptographic Algorithms for Data Security," *International Journal of Computer Applications*, vol. 182, 2020.

[15]. Hermawan, A and Ujianto, E. I. H., (2021). "Implementasi Enkripsi Data Menggunakan Kombinasi AES Dan RSA" *InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan*. Vol 6. No 2. Pp, 325-330.

[16]. Sakshi Parekh and Janak Maru., (2025) "AES, DES, and RSA in Data Security: A Review" *International Journal of Scientific Research and Engineering Development*. Vol 8. No 5. Pp. 1063-1069