International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online))Volume 15, Issue 5, May 2026)

Performance-Based Comparative Study of Symmetric and Asymmetric Encryption Algorithms

Dr. Binita Thakkar

Assistant Professor, VIVA College of Arts, Commerce and Science. Virar, Maharashtra

***Abstract*— Cryptography is an important part of securing digital communication because it helps protect data privacy, maintain information accuracy, and ensure that only authorized users can access sensitive information. As data sharing over networks continues to increase, the performance of cryptographic algorithms has become a major concern, especially in terms of execution speed and memory usage. This study presents the implementation of six commonly used cryptographic algorithms DES, AES, Blowfish, RC4, RC5, and RSA and compares their performance based on execution time and memory consumption. The research evaluates symmetric algorithms such as DES, AES, Blowfish, RC4, and RC5 alongside the asymmetric RSA algorithm to understand the balance between security, speed, and efficiency. The comparative analysis helps in understanding the strengths and limitations of each algorithm and provides useful guidance for selecting suitable encryption techniques for applications such as secure communication, data storage, and key management.**

***Keywords*—cryptography, symmetric algorithms, asymmetric algorithms, python implementation, performance analysis**

I. INTRODUCTION

Cryptography has become one of the most important technologies for ensuring secure communication in the modern digital world. As internet usage, cloud computing, and online transactions continue to grow, protecting sensitive information from unauthorized access has become a major concern for individuals, organizations, and governments. Cryptographic techniques help secure data by converting readable information into an encrypted form that can only be accessed by authorized users. Today, cryptography is widely used in applications such as online banking, e-commerce, email security, healthcare systems, military communication, and social media platforms. It not only protects confidentiality but also ensures data integrity, authentication, and non-repudiation, making it a fundamental component of modern cybersecurity systems. As cyber threats and attacks continue to evolve, the importance of efficient and reliable cryptographic algorithms has increased significantly in both academic research and practical applications.

Cryptographic algorithms are generally classified into two main categories: symmetric key algorithms and asymmetric key algorithms. Symmetric key algorithms use the same secret key for both encryption and decryption, making them faster and more efficient for processing large amounts of data. Due to their high speed and lower computational requirements, these algorithms are commonly used in real-time communication systems, file encryption, and secure data storage. Popular symmetric algorithms such as DES, AES, Blowfish, RC4, and RC5 have been widely studied for their security and performance characteristics. In contrast, asymmetric algorithms use two separate keys, known as public and private keys, to perform encryption and decryption. This approach improves security and simplifies key distribution, making asymmetric algorithms highly suitable for secure key exchange and digital signatures. However, the additional mathematical complexity involved in asymmetric encryption results in higher computational overhead and slower performance compared to symmetric methods.

With the rapid increase in digital communication and data transmission, the demand for secure and efficient cryptographic systems has grown considerably. In many modern applications, especially real-time systems, the performance of encryption algorithms plays a critical role in maintaining speed, reliability, and user experience. Factors such as execution time, memory consumption, and computational efficiency directly affect the practicality of implementing cryptographic techniques in devices with limited resources. Therefore, evaluating the performance of different cryptographic algorithms has become an essential area of research.

This research focuses on the implementation and comparative analysis of six widely used cryptographic algorithms: DES, AES, Blowfish, RC4, RC5, and RSA. The study evaluates these algorithms based on important performance parameters such as execution time and memory usage in order to understand their efficiency and suitability for different applications.

Symmetric algorithms including DES, AES, Blowfish, RC4, and RC5 are compared with the asymmetric RSA algorithm to identify the trade-offs between security strength, processing speed, and resource consumption. The analysis helps in determining which algorithms are more suitable for applications such as secure communication, data encryption, key management, and real-time processing systems. By providing a detailed comparison of these algorithms, the study aims to assist researchers, developers, and industry professionals in selecting appropriate cryptographic techniques for building secure and efficient digital systems.

II. LITERATURE REVIEW

R. Fadlan et al. [1] presents a detailed comparison between AES and Blowfish encryption techniques using files of different sizes. The study shows that AES performs better in both encryption and decryption speed, especially when processing large files, mainly because of its optimized design and hardware support. Blowfish, however, remains effective for smaller files and lightweight systems. The paper concludes that AES is more suitable for high-performance applications, while Blowfish still has value in certain legacy environments.

B. Buhari et al. [2] compares AES, 3DES, and Blowfish in terms of security and performance. Experimental results reveal that Blowfish achieves the fastest execution speed and highest throughput for both small and large files. AES offers a good balance between speed and strong protection against attacks. In contrast, 3DES performs more slowly and provides lower throughput, though it may still be useful in systems with limited resources. The study concludes that AES is the most secure, Blowfish is the most efficient, and 3DES is mainly suitable for older systems.

B. Thakkar [3] provides an in-depth analysis of classical cryptographic techniques such as Caesar, Vigenere, Rail Fence, Columnar, and Vernam ciphers. The paper explains their working mechanisms, strengths, and limitations through Java-based implementations that measure execution time and memory usage. Results indicate that substitution ciphers are generally faster and consume less memory, while transposition ciphers require more resources but offer stronger security. The study highlights the importance of classical cryptography as the basis for many modern encryption methods.

U. Shaikh et al. [4] presents a comparative review of symmetric and asymmetric encryption algorithms including DES, 3DES, AES, RSA, ElGamal, and ECC. The study identifies AES as more efficient than DES among symmetric algorithms, while RSA demonstrates strong encryption and decryption capabilities despite slower key generation. Experimental findings show that symmetric algorithms are faster, whereas asymmetric algorithms provide stronger security features. The paper emphasizes the importance of balancing efficiency and security when selecting encryption techniques.

R. Akter et al. [5] proposes a hybrid encryption model that combines AES-128 and RSA along with HMAC for authentication and integrity in cloud computing. The study explains that AES alone provides high speed, while RSA offers stronger security but slower performance. By combining both methods, the hybrid model achieves improved security while maintaining acceptable performance. Experimental results confirm that the approach enhances confidentiality and authenticity compared to using AES or RSA separately. The paper concludes that hybrid cryptography is an effective solution for cloud data protection.

Y. Salami, E. Zeinali and V. Khajehvand [6] reviews several existing cryptographic algorithms and discusses their advantages, limitations, and current challenges. The study highlights that widely used algorithms such as AES, RSA, and Blowfish each contain certain vulnerabilities that attackers may exploit. It also examines issues related to scalability, performance trade-offs, and resistance to modern cyber threats. The paper stresses the need for continued research into advanced and hybrid cryptographic techniques to improve future security systems.

A. Olutola and M. Olumuyiwa [7] compares AES and RSA based on encryption time, decryption time, key size, and cipher text length. The results show that AES is more efficient because it requires less processing time and produces shorter keys and cipher texts. RSA, although highly secure, consumes more computational resources. The study recommends AES for real-time applications and mobile security because of its speed and practical implementation.

Y. Alemami et al. [8] examines security issues in cloud computing by comparing AES, DES, Blowfish, RSA, and IDEA.

The study finds AES to be the fastest and most effective algorithm for encrypting large amounts of cloud data, while Blowfish is recognized for low memory consumption. RSA and IDEA are considered less efficient and less secure compared to AES and Blowfish. The paper concludes that AES provides the best balance of speed, security, and efficiency for cloud-based systems.

R. Sood and H. Kaur [9] reviews the design, strengths, and weaknesses of RSA, DES, and AES encryption algorithms. The study explains that RSA is slower because it is an asymmetric algorithm, while DES is outdated and vulnerable despite its faster execution speed. AES stands out as the most effective algorithm because of its strong security, high speed, and excellent avalanche effect. The paper concludes that AES is the preferred choice for modern applications and suggests hybrid encryption models for future improvements.

B. Thakkar and B. Thankachan [10] proposed a file-level deduplication method to prevent duplicate uploads in cloud storage systems. The technique uses the MD5 hashing algorithm to verify whether a file already exists before uploading it, which helps save storage space and improve system efficiency. Implemented in Java and tested on text files, the model successfully avoided duplicate uploads. The study also suggests extending the approach using stronger hash algorithms such as SHA-1 or SHA-512 in future work.

B. Thakkar and B. Thankachan [11] introduced a multilevel encryption method that combines transposition ciphers, DES, and Blowfish to improve cloud data security. Although the layered approach increases encryption and decryption time compared to individual algorithms, it significantly strengthens protection by making attacks more difficult. The study also notes that the multilevel method requires less memory than DES and Blowfish in some test cases. The paper concludes that multiple encryption layers improve confidentiality and resilience, with future possibilities of integrating AES or IDEA.

B. Thakkar and B. Thankachan [12] presented a multilevel encryption technique using transposition ciphers to secure cloud data. The method combines Rail Fence and Simple Columnar ciphers sequentially to provide double-layer encryption. The study explains that while single encryption methods offer basic protection, combining multiple techniques significantly improves confidentiality and resistance to attacks.

The paper concludes that multilevel encryption is a more reliable approach for protecting sensitive information in cloud environments.

B. Thakkar and B. Thankachan [13] surveys and compares several encryption algorithms used in cloud security, including AES, DES, Blowfish, RSA, and IDEA. The study highlights that symmetric algorithms generally provide better efficiency. AES offers strong security with fast execution, while Blowfish performs particularly well in memory utilization. RSA, although secure, is slower and requires more memory, whereas DES is outdated and less secure. The paper concludes that AES and Blowfish are the most suitable choices for cloud environments because they balance speed, security, and resource efficiency.

M. Al-Shabi [14] analyzes the performance, strengths, and weaknesses of both symmetric and asymmetric cryptographic algorithms. Symmetric algorithms such as AES, DES, and Blowfish are described as faster and more efficient because they use a shared secret key. In contrast, asymmetric algorithms like RSA, Diffie-Hellman, and ECC provide stronger security through public and private key mechanisms, though they require more computational resources. The paper identifies AES as secure and efficient, Blowfish as lightweight and fast, and RSA/ECC as highly reliable for secure communication.

P. Santoso et al. [15] provides a systematic review of symmetric and asymmetric encryption methods. The study explains that symmetric algorithms such as AES, DES, and Blowfish are faster and more suitable for real-time systems, although they face challenges related to key management. Asymmetric algorithms including RSA, DSA, and Diffie-Hellman offer stronger security through separate public and private keys but operate more slowly and require greater computational power. The paper concludes that symmetric methods are preferred for speed and efficiency, while asymmetric methods are important for secure key exchange and authentication.

M. Yassein et al. [16] presents a detailed study of symmetric and asymmetric encryption algorithms in cloud computing. The paper reviews symmetric methods such as AES, DES, 3DES, and Blowfish, emphasizing their speed and efficiency while also noting weaknesses like DES's vulnerability to brute-force attacks. It also discusses asymmetric algorithms including RSA, Diffie-Hellman, DSA, and ECC, which provide stronger security and better key management but require more processing power.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online))Volume 15, Issue 5, May 2026)

The study concludes that symmetric algorithms are more suitable for handling large-scale data quickly, whereas asymmetric algorithms are essential for secure authentication and key exchange.

N. Garg and P. Yadav [17] compared RSA, Elliptic Curve Cryptography (ECC), and OAEP with respect to secure data transmission. The paper explains that RSA provides strong security but suffers from slower key generation and encryption speed. ECC achieves similar security levels using smaller key sizes, making it faster and more efficient, although slightly more complex to implement. OAEP improves RSA security by adding randomness and padding, thereby increasing resistance against chosen ciphertext attacks. The study concludes that ECC and OAEP provide better efficiency and stronger security than traditional RSA.

S. Chandra et al. [18] reviews symmetric and asymmetric encryption algorithms by comparing their strengths, weaknesses, and applications. Symmetric algorithms such as AES, DES, 3DES, and Blowfish are recognized for their speed and efficiency, though they face challenges related to secure key distribution. Asymmetric algorithms including RSA, Diffie-Hellman, ECC, and DSA provide stronger authentication and secure key exchange but are slower and require more computational resources. The paper concludes that symmetric algorithms are more suitable for high-speed and large-scale data processing, while asymmetric algorithms are essential for secure communication and digital signatures.

III. SYMMETRIC ALGORITHMS

Symmetric key algorithms are one of the most commonly used methods in cryptography for securing digital data. In this approach, the same secret key is used for both encryption and decryption, which means the sender and receiver must share the same key securely before communication begins. These algorithms are known for their high speed, efficiency, and low computational requirements, making them highly suitable for encrypting large amounts of data in real-time applications such as secure messaging, online transactions, cloud storage, multimedia streaming, and file protection systems. Symmetric algorithms are mainly divided into two categories: block ciphers and stream ciphers. Block ciphers, such as DES, AES, Blowfish, and RC5, encrypt data in fixed-size blocks, while stream ciphers like RC4 encrypt data one bit or byte at a time. Due to their faster processing speed and lower memory usage, symmetric algorithms are widely preferred in systems where performance and quick execution are important. However, one major challenge associated with symmetric encryption is secure key distribution, as both communicating parties must safely exchange and manage the secret key, which becomes difficult in large-scale or open network environments.

A. DES

DES (Data Encryption Standard) is one of the earliest symmetric encryption algorithms developed for securing digital information. It was introduced in the 1970s and works by encrypting data in 64-bit blocks using a 56-bit secret key. The algorithm applies multiple rounds of substitutions and rearrangements to convert readable data into encrypted form. DES played an important role in the early development of secure communication systems and was widely used in banking, government, and commercial applications. However, due to advances in computing power, its small key size is no longer considered secure because attackers can break it using brute-force methods. Even though DES is outdated today, it remains historically important because it influenced the development of modern encryption standards such as AES.

B. AES

AES (Advanced Encryption Standard) is one of the most widely used and trusted encryption algorithms in the world today. It was developed to replace DES and provides much stronger security. AES encrypts data in 128-bit blocks and supports key sizes of 128, 192, and 256 bits, allowing different levels of security. The algorithm performs several rounds of substitution, shifting, and mixing operations to protect the data from unauthorized access. AES is known for its strong security, fast performance, and efficient memory usage, making it suitable for a wide range of applications including secure communication, online banking, wireless networks, cloud storage, and government systems. Due to its reliability and resistance to attacks, AES is considered the global standard for modern encryption.

C. Blowfish

Blowfish is a symmetric block cipher developed by Bruce Schneier in 1993 as a fast and flexible encryption algorithm. It encrypts data in 64-bit blocks and supports variable key lengths ranging from 32 bits to 448 bits, allowing users to choose different levels of security based on their requirements.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online))Volume 15, Issue 5, May 2026)

Blowfish uses a 16-round encryption process that provides both speed and security, especially in software-based applications. It became popular because of its efficiency and free availability for public use. Blowfish is still found in some legacy systems, password management tools, and embedded devices because of its lightweight design and reliable performance.

D. RC4

RC4 is a symmetric stream cipher developed by Ron Rivest in 1987. Unlike block ciphers, RC4 encrypts data one byte at a time by generating a pseudorandom keystream that is combined with the plaintext using XOR operations. The algorithm became very popular because of its simplicity, high speed, and low memory requirements. It was widely used in security protocols such as SSL/TLS and wireless encryption systems like WEP. However, over time, researchers discovered several weaknesses in RC4, especially in its key scheduling process and keystream generation, making it vulnerable to attacks. As a result, RC4 is no longer recommended for secure applications. Despite its security issues, RC4 remains important for understanding the basic concepts of stream cipher encryption.

E. RC5

RC5 is a symmetric block cipher introduced by Ron Rivest in 1994. It is known for its simple structure and flexibility because users can adjust parameters such as word size, number of rounds, and key length according to their security and performance needs. RC5 uses operations like XOR, modular addition, and data-dependent rotations to encrypt data efficiently. Its adaptable design allows it to perform well in different hardware and software environments. RC5 gained attention for its efficiency and customizable nature, making it suitable for various encryption applications. Although it is less commonly used today compared to AES, RC5 remains an important algorithm in cryptographic research and education due to its innovative design principles.

IV. ASYMMETRIC ALGORITHMS

Asymmetric key algorithms, also known as public-key cryptography, provide a more secure approach to communication by using two different keys: a public key and a private key. The public key is openly shared and used for encryption, while the private key remains confidential and is used for decryption.

This method removes the need for securely sharing a single secret key between users, making asymmetric cryptography highly effective for secure communication over public networks such as the internet. Algorithms such as RSA are widely used for applications including secure key exchange, digital signatures, authentication, and secure communication protocols like SSL/TLS and encrypted email systems. Although asymmetric algorithms offer stronger security and better scalability compared to symmetric methods, they involve more complex mathematical computations, resulting in higher execution time and greater memory usage. Because of their slower performance, asymmetric algorithms are generally not used for encrypting large volumes of data directly. Instead, they are commonly combined with symmetric algorithms, where asymmetric encryption is used for secure key exchange and symmetric encryption is used for faster data transmission.

A. RSA

RSA is one of the most well-known asymmetric encryption algorithms and was introduced in 1978. Unlike symmetric algorithms, RSA uses two keys: a public key for encryption and a private key for decryption. This approach allows secure communication without the need to share a secret key between users. RSA is based on complex mathematical calculations involving large prime numbers, which makes it highly secure. It is commonly used for secure key exchange, digital signatures, authentication, and internet security protocols such as SSL/TLS. However, RSA requires more computational power and memory compared to symmetric algorithms, making it slower for encrypting large amounts of data. For this reason, RSA is mainly used for security-related tasks such as authentication and secure key management rather than bulk data encryption.

V. IMPLEMENTATION

All the above cryptographic techniques were implemented using Python to analyze their performance during execution. During the implementation process, the total encryption and decryption time, referred to as execution time, was measured in milliseconds. In addition, the amount of memory consumed during execution was calculated in bytes. Table I presents the comparative analysis of the execution time and memory usage of the various algorithms.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online))Volume 15, Issue 5, May 2026)

TABLE I
ALGORITHMS EXECUTION TIME AND MEMORY UTILIZATION

Algorithms	Execution Time (msec)	Memory Used (kB)
DES	1.07	150.5
AES	0.8	4.32
Blowfish	1.27	148.5
RC4	0.94	6.51
RC5	5.58	3.23
RSA	31.47	173.81

Based upon the implementation of various algorithms, analysis of total execution time is shown in Fig 1 and memory utilization is shown in Fig. 2.

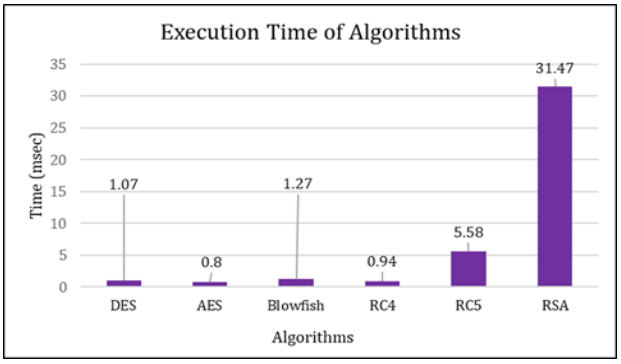

Figure 1. Execution time of algorithms

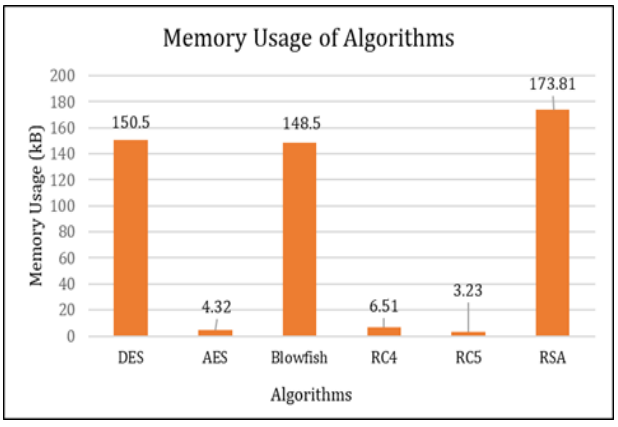

Figure 2. Memory usage of algorithms

VI. COMPARATIVE ANALYSIS OF ALGORITHMS

Table II presents a comparative analysis of various algorithms based on parameters such as strengths, weakness and use.

TABLE III
COMPARATIVE ANALYSIS OF VARIOUS ALGORITHMS

Techniques	Strength	Weakness	Use
DES	Simple design, historically important, fast in hardware	Small 56-bit key size makes it vulnerable to brute-force attacks; considered obsolete	Legacy systems, academic demonstrations
AES	Strong security, flexible key sizes (128/192/256 bits), efficient in both hardware and software	Slightly slower than lightweight ciphers in constrained devices	Modern encryption standard, secure communications, data storage
Blowfish	Variable key length (32–448 bits), fast and efficient, good for embedded systems	64-bit block size limits security in large data sets; largely replaced by AES	Legacy applications, password hashing, embedded devices
RC4	Extremely fast, lightweight, low memory usage	Weak key scheduling, biased keystream, insecure in modern contexts	Historical use in SSL/TLS, WEP; educational purposes
RC5	Highly flexible (variable word size, rounds, key length), efficient operations (XOR, rotations)	Less widely adopted, higher execution time compared to AES	Academic research, adaptable cryptographic experiments
RSA	Strong security for key exchange and digital signatures,	Computationally expensive, slower than symmetric algorithms, high memory usage	Secure key exchange, digital signatures, SSL/TLS, secure email

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online))Volume 15, Issue 5, May 2026)

	widely adopted		

VII. CONCLUSION

This study focused on the implementation and performance analysis of six widely used cryptographic algorithms—DES, AES, Blowfish, RC4, RC5, and RSA—using Python programming. The algorithms were evaluated based on two important performance factors: execution time and memory usage, which play a major role in determining their suitability for practical applications. The results revealed that symmetric algorithms generally perform faster and consume less memory compared to the asymmetric RSA algorithm. Among all the techniques tested, AES showed the fastest execution speed and RC5 showed the lowest memory consumption, making it highly efficient for lightweight applications. AES also demonstrated the best balance between speed, memory efficiency, and strong security, which explains why it is widely accepted as the global standard for data encryption. Blowfish and RC5 also provided good performance and flexibility, although RC5 showed comparatively higher execution time, making it less suitable for time-sensitive applications. On the other hand, RSA required significantly more computational resources and memory, but it remains highly important for secure key exchange, authentication, and digital signatures. The study highlights that the choice of a cryptographic algorithm should depend on the specific needs of the application. Lightweight symmetric algorithms are more suitable for real-time communication and resource-limited systems, while RSA is preferred for secure key management. This research provides useful insights for developers, researchers, and students working in the field of cybersecurity and secure system design.

References

[1] R. Fadlan, F. Siregar, N. Dly, S. Wulandari, N. A. Siregar, and I. Rusydi, "COMPARATIVE ANALYSIS OF ENCRYPTION AND DECRYPTION SPEED OF AES AND BLOWFISH ALGORITHMS," *Interdiscip. J. Glob. Multidiscip.*, vol. 2, no. 1, pp. 449–458, 2026.

[2] B. A. Buhari *et al.*, "Performance and Security Analysis of Symmetric Data Encryption Algorithms: AES, 3DES and Blowfish," *Int. J. Adv. Netw. Appl.*, vol. 16, no. 04, pp. 6473–6486, 2025, doi: 10.35444/ijana.2025.16404.

[3] B. Thakkar, "A Comprehensive Study of Substitution and Transposition Techniques in Cryptographic Systems," *Int. J. Comput. Sci. Trends Technol.*, vol. 13, no. 2, pp. 15–19, 2025.

[4] U. Shaikh, A. Mirza, S. Layad, M. Razi, M. Shaikh, and Ish, "A Comparative Survey of Symmetric and Asymmetric Key Cryptography Algorithms," *2nd Int. Multidiscip. Conf. Emerg. Trends Eng. Technol. (2nd IMCEET-2024)*, pp. 257–262, 2024.

[5] R. Akter, M. A. R. Khan, F. Rahman, S. J. Soheli, and N. J. Suha, "RSA and AES Based Hybrid Encryption Technique for Enhancing Data Security in Cloud Computing," *Int. J. Comput. Appl. Math. Comput. Sci.*, vol. 3, pp. 60–71, 2023, doi: 10.37394/232028.2023.3.8.

[6] Y. Salami, E. Zeinali, and V. Khajehvand, "Cryptographic Algorithms: A Review of the Literature, Weaknesses and Open Challenges," *J. Comput. Robot.*, vol. 16, no. 2, pp. 63–115, 2023, doi: 10.22094/JCR.2023.1983496.1298.

[7] A. Olutola and M. Olumuyiwa, "Comparative Analysis of Encryption Algorithms," *Eur. J. Technol.*, vol. 7, no. 1, pp. 1–9, 2023, doi: 10.47672/ejt.1312.

[8] Y. Alemami, A. M. Al-Ghonmein, K. G. Al-Moghrabi, and M. A. Mohamed, "Cloud data security and various cryptographic algorithms," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, pp. 1867–1879, 2023, doi: 10.11591/ijece.v13i2.pp1867-1879.

[9] R. Sood and H. Kaur, "A Literature Review on RSA, DES and AES Encryption Algorithms," *Emerg. Trends Eng. Manag.*, pp. 57–63, 2023, doi: 10.56155/978-81-955020-3-5-07.

[10] B. Thakkar and B. Thankachan, "A Data Deduplication Approach for Eliminating Duplicate File Upload over Cloud," *Int. J. Enhanc. Res. Sci. Technol. Eng.*, vol. 11, no. 2, pp. 13–17, 2022.

[11] B. Thakkar and B. Thankachan, "An Approach for Enhancing Security of Data over Cloud Using Multilevel Algorithm," in *Congress on Intelligent Systems, Lecture Notes on Data Engineering and Communications Technologies*, Springer Nature Singapore Pte Ltd, 2022, pp. 305–318. doi: https://doi.org/10.1007/978-981-16-9416-5_22.

[12] B. Thakkar and B. Thankachan, "A Multilevel Approach of Transposition Ciphers for Data Security over Cloud," *GIS Sci. J.*, vol. 8, no. 5, pp. 1732–1738, 2021.

[13] B. Thakkar and B. Thankachan, "A Survey for Comparative Analysis of various Cryptographic Algorithms used to Secure Data on Cloud," *Int. J. Eng. Res. Technol.*, vol. V9, no. 08, pp. 753–756, 2020, doi: 10.17577/ijertv9is080328.

[14] M. A. Al-Shabi, "A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security," *Int. J. Sci. Res. Publ.*, vol. 9, no. 3, pp. 576–589, 2019, doi: 10.29322/IJSRP.X.X.2018.pXXXX.

[15] P. P. Santoso *et al.*, "Systematic literature review: Comparison study of symmetric key and asymmetric key algorithm," in *IOP Conference Series: Materials Science and Engineering*, 2018, pp. 1–6. doi: 10.1088/1757-899X/420/1/012111.

[16] M. B. Yassein, S. Aljawarneh, E. Qawasmeh, W. Mardini, and Y. Khamayseh, "Comprehensive study of symmetric key and asymmetric key encryption algorithms," in *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, 2017, pp. 1–7. doi: 10.1109/ICEngTechnol.2017.8308215.

[17] N. Garg and P. Yadav, "Comparison of Asymmetric Algorithms in Cryptography," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 4, pp. 1190–1196, 2014, [Online]. Available: www.ijcsmc.com

[18] S. Chandra, S. Paira, S. S. Alam, and G. Sanyal, "A comparative survey of symmetric and asymmetric key cryptography," in *2014 International Conference on Electronics, Communication and Computational Engineering, ICECCE 2014*, 2014, pp. 83–93. doi: 10.1109/ICECCE.2014.7086640.