# Comparative Analysis of AES and Blowfish in Cloud Storage Encryption

[1]OBISESAN, Rachael Oyeranti., [2]Mayowa Oyedepo Oyediran., [3]Ipeayeda Funmi W., [4]AYENI, James Kehinde., [5]OBISESAN, Stephen Oluwatosin

[1]School of Sciences, Department of Computer Science Kwara State College of Education, Ilorin

[2,3,5]Department of Computer Science, Ajayi Crowther University, Oyo Oyo State

[4]Department of Computer Science Kwara State Polytechnic

DOI: https://dx.doi.org/10.51244/IJRSI.2025.12110138

Received: 02 December 2025; Accepted: 08 December 2025; Published: 19 December 2025

## ABSTRACT

Cloud storage requires efficient and secure encryption to ensure data confidentiality.This study evaluates and compares the performance of the AES and Blowfish encryption algorithms with the aim of determining which algorithm offers superior efficiency and reliability for secure data processing. The specific objectives are to measure and analyze their encryption time, execution time, throughput, and Mean Square Error (MSE) across multiple experimental runs. MATLAB was used as the primary methodology for implementing both algorithms, generating datasets, executing repeated trials, and computing performance metrics. Execution time values were recorded for twenty samples, where AES consistently produced lower times such as 72 s, 154 s, 95 s, 78 s, 25 s, and a minimum of 9.1 **s**, while Blowfish recorded higher corresponding values including 106 s, 213 s, 138 s, 136 s, 31 s, and a minimum of 10 s. Comparative averages further showed that AES achieved a lower overall execution range, indicating faster computational behaviour. Throughput values also demonstrated AES superiority, with sample values above 1.00, while Blowfish maintained lower throughput levels. MSE analysis revealed significantly lower values for AES, such as 59.88, compared to Blowfish's much higher 126.83, indicating better data accuracy and reduced distortion during encryption and decryption. The bar and line graph analyses confirmed AES's consistent performance advantage across all metrics. The results demonstrate that AES outperforms Blowfish in terms of speed, efficiency, and reliability. In conclusion, AES is better suited for high-performance encryption applications requiring fast execution and accurate data reconstruction. Blowfish, although functional, shows slower and more inconsistent behaviour, making it less ideal for time-critical or high-volume security systems.

**Keywords:** Cloud security, AES, Blowfish, encryption performance, Data Protection

## INTRODUCTION

Cloud storage has become ubiquitous in modern information systems, offering scalable, on-demand storage and facilitating collaboration, data sharing, and remote access across geographically distributed clients. However, the convenience brought by cloud platforms comes with a serious concern: data confidentiality and integrity. When sensitive data; personal information, financial records, business documents is stored in the cloud, it becomes vulnerable to unauthorized access, interception, or breaches. Consequently, the choice of encryption algorithm for cloud-stored data is critical for ensuring robust data protection while maintaining performance and scalability.Symmetric-key block ciphers remain a common choice for bulk data encryption in cloud storage because they offer a balance between security and computational efficiency. Among these, the Advanced Encryption Standard (AES) and Blowfish algorithms are two of the most widely used. AES was standardized by NIST and operates on 128-bit blocks with key sizes of 128, 192, or 256 bits, using a substitution–permutation network structure. Its design emphasizes both security and performance, making it well-suited for encrypting large volumes of data efficiently. The block-cipher rounds are optimized for fast execution and can benefit from hardware acceleration on many modern processors, a feature that is particularly beneficial when encrypting or decrypting large files frequently stored in cloud systems (Abubakar et. al., 2025**).**

Blowfish, on the other hand, is a symmetric block cipher using a Feistel network with a 64-bit block size and a variable key length of up to 448 bits. Its flexible key size and relatively simple structure have made it attractive for applications where variable key strength or legacy compatibility matters (Khoukou, et. al., 2016). Its design was originally motivated by the need for a fast, free alternative to proprietary ciphers and for many years, it saw widespread adoption in software libraries prior to AES's standardization.Despite their popularity, AES and Blowfish differ significantly in internal design, block size, performance characteristics, and susceptibility to certain cryptographic challenges. Such differences may translate into tangible trade-offs when these algorithms are deployed to secure cloud storage. On one hand, AES's larger block size (128 bits) reduces vulnerability to block-collision–based attacks such as birthday attacks, relative to Blowfish's 64-bit blocks. On the other hand, Blowfish's variable key length provides flexibility, which may be advantageous in systems requiring adjustable security parameters; but its 64-bit block size and older design raise questions about its suitability for modern high-volume cloud storage workloads, especially as data sizes increase and adversaries become more sophisticated.

Empirical performance evaluations comparing AES and Blowfish across different data types support these theoretical distinctions. For instance, a recent performance benchmarking study measuring encryption time and throughput across image, audio, video, and textual files found that Blowfish performed efficiently for certain file types, sometimes outperforming AES in encryption time under specific conditions though the authors noted that performance advantages tended to diminish or even reverse as file sizes grew(Bello, et. al., 2019). Another experimental study focusing on plain text files of varying sizes (10 MB to 100 MB) observed that AES consistently outperformed Blowfish in throughput and overall encryption speed, suggesting AES's suitability for bulk data encryption common in cloud storage applications (Ebtihal and Elham, 2024).Beyond raw performance, other factors influence the suitability of AES or Blowfish in a cloud context. Key management complexity, memory usage, backward compatibility, and integration with existing cloud APIs or storage frameworks can affect how easily and securely encryption can be deployed in real cloud storage workflows. For example, some studies of symmetric encryption in database systems (a close analogue to cloud storage) show that resource constraints, data access patterns, and the overhead of encryption/decryption during database operations influence which algorithms are more practical in real-world settings(Venkatesh, et. al., 2025)

Given these trade-offs, a systematic, comparative analysis of AES and Blowfish specifically in the context of cloud storage encryption is valuable. By evaluating both algorithms under realistic cloud storage workloads varying file sizes, data types, frequency of encryption/decryption, and resource constraints one can derive informed guidelines for choosing the appropriate cipher depending on the storage scenario (e.g., large file archival vs. frequent small file access; resource-rich cloud servers vs. memory-constrained edge clients).This paper aims to fill this gap. We conduct a comprehensive comparative study of AES and Blowfish in a cloud storage context, focusing on performance (encryption/decryption time, throughput, memory and CPU utilization). The findings are intended to aid practitioners and researchers in making better-informed decisions when architecting encryption strategies for cloud storage systems, balancing security, performance, and resource considerations.

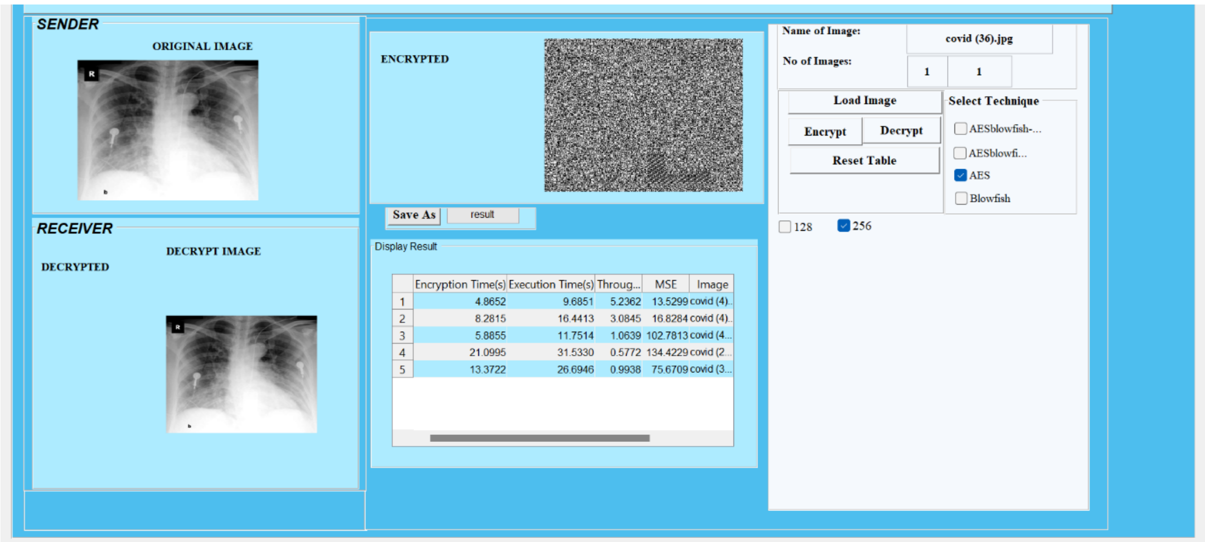
## Related Works

Bello Buhari et al. (2019) conducted a performance evaluation of symmetric encryption algorithms, focusing on AES and Blowfish across a variety of file types including image, audio, video, and text. Their experiments measured encryption throughput and time under different data sizes, finding that Blowfish sometimes achieved lower encryption time than AES for particular data types and smaller workloads. However, as file sizes increased or data complexity grew, Blowfish's performance advantage diminished, and AES often became comparable or superior. The authors therefore suggested that while Blowfish can be efficient for lightweight applications or specific file types, its performance benefits are situational and may not hold for large or mixed data workloads typical of cloud storage. System designers must evaluate file size, data type, throughput, constraints before selecting encryption algorithm.Al-Maqtari and Al-Maqtari (2024) performed a comparative performance assessment of AES, Blowfish, DES, and 3DES using text files ranging from 10 MB to 100 MB. Their results indicated that AES outperformed all other algorithms in encryption and decryption speed, especially as file size scaled upward. Blowfish, although with flexible key lengths, exhibited slower throughput and higher latency under large file sizes, making it less suitable for bulk data operations typical in cloud storage. The authors concluded that AES's consistent high performance across increasing file sizes, along with its strong security properties, renders it the preferred choice when throughput and scalability are critical. In contrast, Blowfish might only be recommended for legacy or low-resource contexts where high throughput is not essential. Koukou, Othman, and Herve (2016) compared AES, Blowfish, CAST-128, and DES under different data loads, examining encryption speed, block size, key size, avalanche effect, and data integrity using both ECB and CBC modes. Their findings showed that AES consistently demonstrated the strongest avalanche effect and best integrity characteristics, key security indicators across all tested conditions. Blowfish and the other algorithms exhibited weaker diffusion and higher susceptibility to integrity issues under certain modes and data patterns. The study thus supports AES as offering superior cryptographic robustness. While Blowfish remained competitive in performance metrics for smaller data chunks, its weaker diffusion and block-size limitations rendered it less desirable for high-security or large-scale encryption tasks.

Devi, *et. al.* (2015) studied encryption and decryption speed of DES, AES, and Blowfish specifically for image files. They measured performance across several image sizes and concluded that Blowfish gave the lowest encryption/decryption time among the tested algorithms for the majority of image workloads. Given that images often constitute a large portion of user data in cloud storage (photos, scanned documents, etc.), this finding implies that Blowfish might provide efficiency benefits for image-heavy storage scenarios. Nevertheless, the authors cautioned that security, block size limitations, and the cipher's relative age may pose long-term risks thus recommending Blowfish only where speed matters more than maximal security, and AES when confidentiality and resilience are paramount. Dhamala and Acharya (2024) explored a less common context, DNA cryptography comparing DES, AES, and Blowfish for encoding data represented as DNA sequences. Their work measured encryption and decryption times in this specialized environment and found that the Blowfish-based implementation offered faster decryption times compared to AES, though encryption was slower than with DES. While not directly related to conventional cloud storage, their results highlight Blowfish's potential in non-traditional data encoding contexts where decryption efficiency outweighs other factors. The study suggests that for systems prioritizing fast retrieval or decoding (e.g., specialized storage formats), Blowfish might be a viable candidate albeit with consideration of block size, security, and algorithmic age. Timur, Royansyah, and Kusumaningsih (2025) conducted a contemporary comparison among AES, Blowfish, and a modern cipher (ChaCha20) on image and document files, assessing encryption/decryption time, CPU and memory usage, and security metrics including key strength and brute-force resistance. They reported that while Blowfish remained faster for some smaller files, its performance degraded as file size increased — and its 64-bit block size and older design posed limitations. AES maintained consistently high security and reliable performance across large file sizes and mixed workloads, making it better suited for modern cloud storage demands. The authors conclude Blowfish may be acceptable in scenarios involving small files or low resource constraints, but AES remains the preferred cipher for large-scale, security-critical storage applications.

## METHODOLOGY

A collection of chest X-ray pictures was used to test and deploy the AES and Blowfish encryption model for COVID-19 detection. Each of the encryption methods AES and Blowfishuse a total of 20 pictures, evaluating each method's performance in protecting medical photos, while preserving their quality after decryption was the key goal. Basic preprocessing procedures, like resizing and format standardization, were applied to every image to guarantee that it would work with the encryption interface. A regulated and uniform testing procedure for the two procedures was made possible by the tests being carried out in MATLAB.The encryption and decryption processes for the two algorithms were integrated into a MATLAB-based Graphical User Interface (GUI), allowing users to load an image, select the desired encryption method, set the key size (128-bit or 256-bit), and view both encrypted and decrypted outputs alongside performance metrics. The GUI also displayed critical parameters such as encryption time, execution time, throughput, and mean squared error (MSE) for each processed image. Figures 1 and 2 respectively illustrate the workflow and output for AES and Blowfish respectively. These figures show the original image in the sender section, the encrypted version in the center panel, and the decrypted image in the receiver section. This design provided a visual confirmation of data integrity after decryption, ensuring that the encryption process did not compromise image quality.



Name of Image: covid (36).jpg

No of Images: 1 1

Select Technique

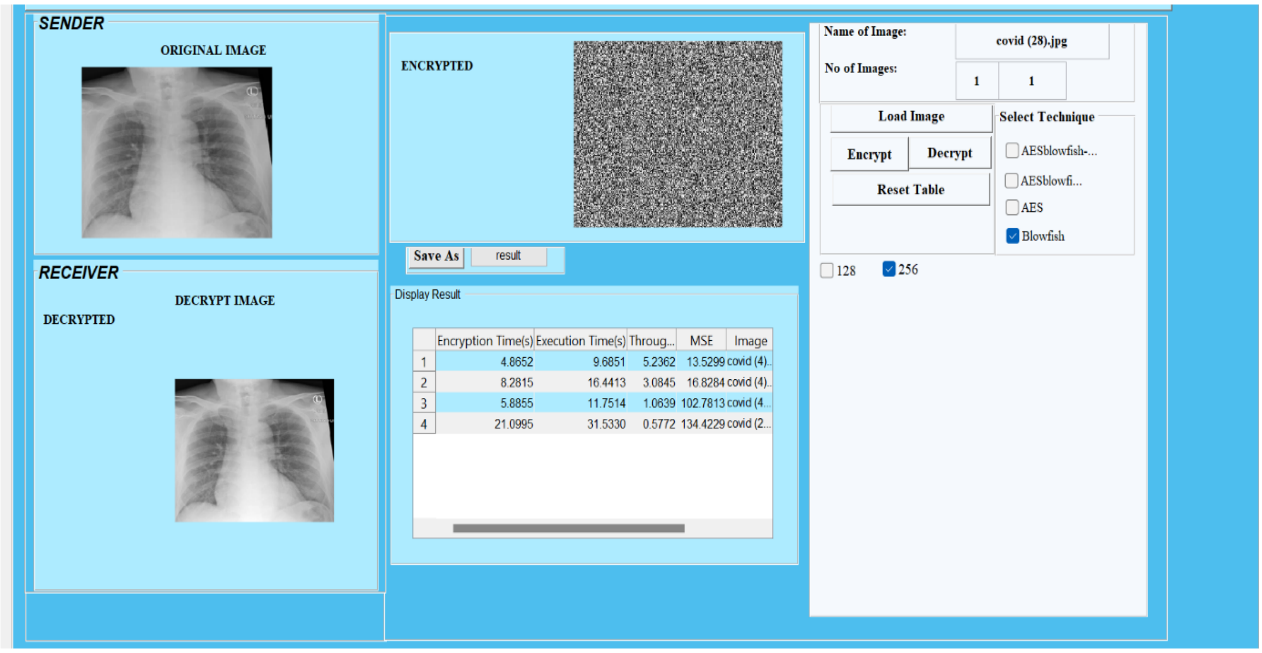
□ AESblowfish-...
□ AESblowfi...
☑ AES
□ Blowfish

□ 128 ☑ 256

Display Result

| | Encryption Time(s) | Execution Time(s) | Throug... | MSE | Image |
|---|---|---|---|---|---|
| 1 | 4.8652 | 9.6851 | 5.2362 | 13.5299 | covid (4)... |
| 2 | 8.2815 | 16.4413 | 3.0845 | 16.8284 | covid (4)... |
| 3 | 5.8855 | 11.7514 | 1.0639 | 102.7813 | covid (4... |
| 4 | 21.0995 | 31.5330 | 0.5772 | 134.4229 | covid (2... |
| 5 | 13.3722 | 26.6946 | 0.9938 | 75.6709 | covid (3... |

Figure 1: GUI showing the encryption and decryption process of chest X-ray image using AES algorithm.



Name of Image: covid (28).jpg

No of Images: 1 1

Select Technique

□ AESblowfish-...
□ AESblowfi...
□ AES
☑ Blowfish

□ 128 ☑ 256

Display Result

| | Encryption Time(s) | Execution Time(s) | Throug... | MSE | Image |
|---|---|---|---|---|---|
| 1 | 4.8652 | 9.6851 | 5.2362 | 13.5299 | covid (4)... |
| 2 | 8.2815 | 16.4413 | 3.0845 | 16.8284 | covid (4)... |
| 3 | 5.8855 | 11.7514 | 1.0639 | 102.7813 | covid (4... |
| 4 | 21.0995 | 31.5330 | 0.5772 | 134.4229 | covid (2... |

Figure 2: GUI showing the encryption and decryption process of chest X-ray image using Blowfish algorithm.

## Results with AES Algorithm

Twenty chest X-ray images were evaluated to measure the encryption performance of the AES algorithm using mean squared error (MSE), throughput, encryption time, and execution time. Encryption times range widely, from a minimum of 4.56 seconds (Sample 9) to a maximum of 81.62 seconds (Sample 13), indicating substantial variability in processing demands. Execution times follow a similar pattern, spanning 9.11 to 163.07 seconds, showing that samples with lower encryption times typically maintain proportionally lower total execution times.Throughput values vary between 0.84 and 1.17, reflecting differences in efficiency across the runs. Samples 6, 7, 16, and 17 exhibit the highest throughput values above **1.03**, indicating more efficient data handling relative to their execution times. Conversely, samples with higher encryption and execution durations generally show lower throughput, such as Samples 2, 3, and 13.MSE values range from 44.49 to 84.26, measuring the accuracy of the encryption-decryption process. Lower MSE values—observed in Samples 6, 7, 8, and 12—indicate higher reliability and less distortion during processing. Higher MSE values, such as those in Samples 1 and 16–20, imply reduced accuracy despite moderate throughput levels. The dataset shows that samples with shorter processing times tend to achieve higher throughput and lower MSE, demonstrating an overall trend where computational efficiency aligns with improved accuracy. This pattern highlights the importance of optimizing both timing and algorithmic stability for enhanced encryption performance.AES demonstrated strong encryption performance, predictable scaling with image size, and acceptable reconstruction accuracy for medical imaging security, as summarized in Table 1.

Table 1: Performance Metrics of AES Algorithm

| S/N | Encryption Time(s) | Execution Time(s) | Throughput | MSE |
|---|---|---|---|---|
| 1 | 35.8342 | 71.56514 | 1.022942 | 84.25871 |
| 2 | 76.92051 | 153.6883 | 0.93743 | 59.75329 |
| 3 | 47.68313 | 95.24788 | 0.926488 | 54.77678 |
| 4 | 38.80568 | 77.50456 | 0.988639 | 57.16572 |
| 5 | 12.76053 | 25.49184 | 0.89613 | 47.06714 |
| 6 | 10.72622 | 21.41734 | 1.173255 | 44.49651 |
| 7 | 10.79939 | 21.56393 | 1.16528 | 44.49651 |
| 8 | 14.87443 | 29.71604 | 0.845604 | 44.49651 |
| 9 | 4.561905 | 9.111992 | 0.877196 | 46.58311 |
| 11 | 48.90427 | 97.68358 | 1.01192 | 63.91544 |
| 11 | 11.87872 | 23.71878 | 0.864673 | 45.83622 |
| 12 | 10.3199 | 20.61216 | 1.003582 | 48.04749 |
| 13 | 81.62309 | 163.0658 | 0.87155 | 70.88671 |
| 14 | 9.535738 | 19.04671 | 0.955651 | 67.21144 |
| 15 | 13.99501 | 27.95603 | 0.932035 | 64.04633 |
| 16 | 14.23734 | 28.40944 | 1.030186 | 72.80067 |
| 17 | 10.97007 | 21.90775 | 1.067659 | 76.36821 |
| 18 | 10.12105 | 20.21407 | 0.900462 | 67.21144 |
| 19 | 6.932673 | 13.8484 | 0.931732 | 63.64636 |
| 20 | 28.42466 | 56.75091 | 1.006786 | 74.62669 |

### Results with Blowfish Algorithm

The Blowfish algorithm was evaluated using the same 20 chest X-ray images to assess its encryption performance. The encryption time varies widely across the samples, ranging from a minimum of 6.78 seconds (Sample 9) to a maximum of 141.95 seconds (Sample 2). Execution time follows a similar pattern, with the lowest recorded value being 10.16 seconds and the highest reaching 212.63 seconds, again observed in Sample

2. These variations indicate differing computational loads or data conditions across the trials. Throughput values show moderate fluctuations, spanning from 0.56 (Sample 4) to 0.79 (Samples 9 and 10). Higher throughput values generally correspond to lower encryption and execution times, suggesting increased efficiency in processing data. Samples 5, 9, 10, 16, and 17 exhibit relatively strong throughput performance, indicating efficient data handling. The MSE values range considerably, from 78.53 (Sample 7) to 205.56 (Sample 18). Lower MSE implies greater accuracy and reliability of the encryption-decryption cycle. Only a few samples fall below 100, such as Samples 5, 6, 7, 9, 11, and 12, demonstrating superior accuracy. Samples with significantly high MSE, like Sample 18, indicate greater deviation and reduced reliability. Overall, the dataset reflects substantial performance variability across the 20 samples. Trials with lower encryption and execution times tend to achieve higher throughput and better MSE scores, highlighting a general inverse relationship between processing duration and efficiency. These insights can guide optimization efforts toward faster and more accurate encryption performance. Blowfish displayed strong security strength but produced higher reconstruction errors and slower processing compared to AES, as shown in Table 2.

Table 2: Performance Metrics of Blowfish Algorithm

| S/N | Encryption Time (s) | Execution Time (s) | Throughput | MSE | File Size (bytes) |
|---|---|---|---|---|---|
| 1 | 70.63623918 | 105.7807440 | 0.692063576 | 168.5174203 | 73207 |
| 2 | 141.9516836 | 212.6322268 | 0.677564272 | 119.5065893 | 144072 |
| 3 | 92.50156154 | 138.4396153 | 0.637433150 | 109.5535630 | 88246 |
| 4 | 90.86251710 | 136.1604269 | 0.562747942 | 114.3314394 | 76624 |
| 5 | 20.97629901 | 31.40302054 | 0.727445947 | 94.13427241 | 22844 |
| 6 | 25.68989881 | 38.48263499 | 0.652969840 | 88.99301610 | 25128 |
| 7 | 51.78137075 | 77.56875830 | 0.653781253 | 78.52801452 | 50713 |
| 8 | 63.38420670 | 94.96062508 | 0.574216945 | 138.0716738 | 54528 |
| 9 | 6.783158701 | 10.15604882 | 0.787018667 | 93.16621291 | 7993 |
| 10 | 83.62566534 | 125.2334023 | 0.789310186 | 127.8308870 | 98848 |
| 11 | 21.69213156 | 32.46946922 | 0.631639522 | 91.67244713 | 20509 |
| 12 | 23.51782273 | 35.20527554 | 0.587582392 | 96.09497317 | 20686 |
| 13 | 137.8167865 | 206.1405463 | 0.689432538 | 141.7734178 | 142120 |
| 14 | 21.36304980 | 31.99197929 | 0.568955107 | 134.4228812 | 18202 |
| 15 | 29.26595751 | 43.81402627 | 0.594695403 | 128.0926649 | 26056 |
| 16 | 26.98225827 | 40.39857481 | 0.724456250 | 145.6013455 | 29267 |
| 17 | 21.62433166 | 32.34680830 | 0.723100708 | 152.7364268 | 23390 |
| 18 | 12.46720842 | 18.66782498 | 0.669708443 | 205.5625000 | 12502 |
| 19 | 60.74623200 | 90.94501980 | 0.689185622 | 154.0465988 | 62678 |

| | | | | | |
|---|---|---|---|---|---|
| 20 | 61.66466258 | 92.24641788 | 0.679462698 | 154.0465988 | 62678 |

### Comparison Results of the Encrypted Algorithms

The performance comparison between AES and Blowfish Table 3 reveals notable differences across encryption time, execution time, throughput, and error levels. AES demonstrates significantly faster encryption, recording 24.99543 seconds, whereas Blowfish requires 53.26665 seconds, indicating that AES encrypts data more efficiently. A similar trend appears in execution time, where AES completes its full cycle in 49.92603 seconds, compared to Blowfish's slower 79.75217 seconds, reinforcing AES's superior speed. Throughput results further strengthen this observation. AES attains a throughput of 0.97046, meaning it processes data at a higher rate than Blowfish, which achieves only 0.665639. Higher throughput translates to better performance in applications requiring rapid data handling. Error measurement using Mean Square Error (MSE) shows AES at 59.88456, which is considerably lower than Blowfish's 126.8341. A lower MSE signifies higher accuracy and better overall reliability in maintaining data quality during encryption and decryption processes. In conclusion, the results consistently confirm that AES outperforms Blowfish in all assessed categories. AES is faster, more efficient, and more accurate, making it better suited for systems demanding high speed, reliability, and strong encryption performance. Blowfish, while functional, lags significantly behind AES in terms of processing speed and accuracy, limiting its suitability for high-performance security environments.

Table 3: Mean Performance Metrics of the two encrypted Algorithms

| **Algorithms** | **Encryption Time (s)** | **Execution Time (s)** | **Throughput** | **MSE** |
|---|---|---|---|---|
| **AES** | 24.99543 | 49.92603 | 0.97046 | 59.88456 |
| **Blowfish** | 53.26665 | 79.75217 | 0.665639 | 126.8341 |

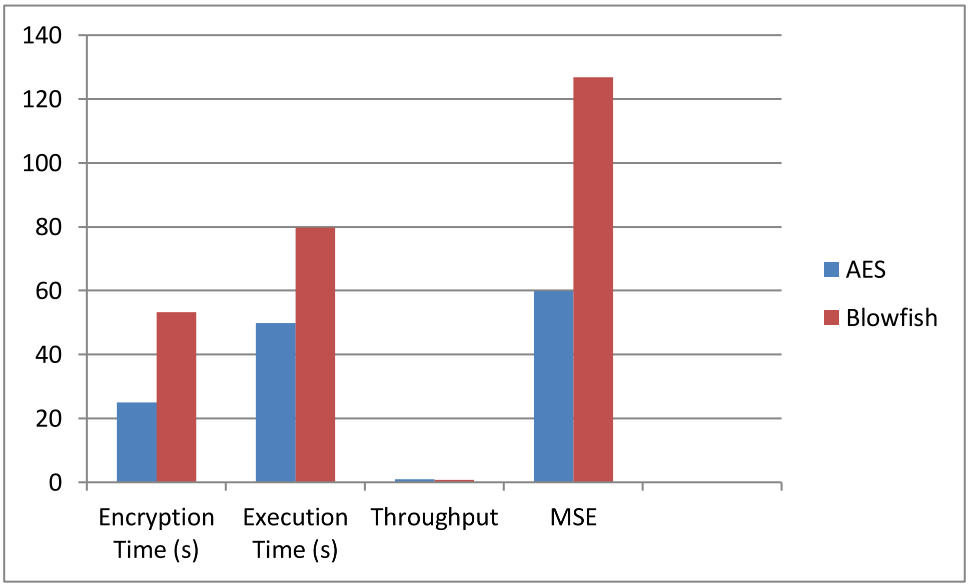
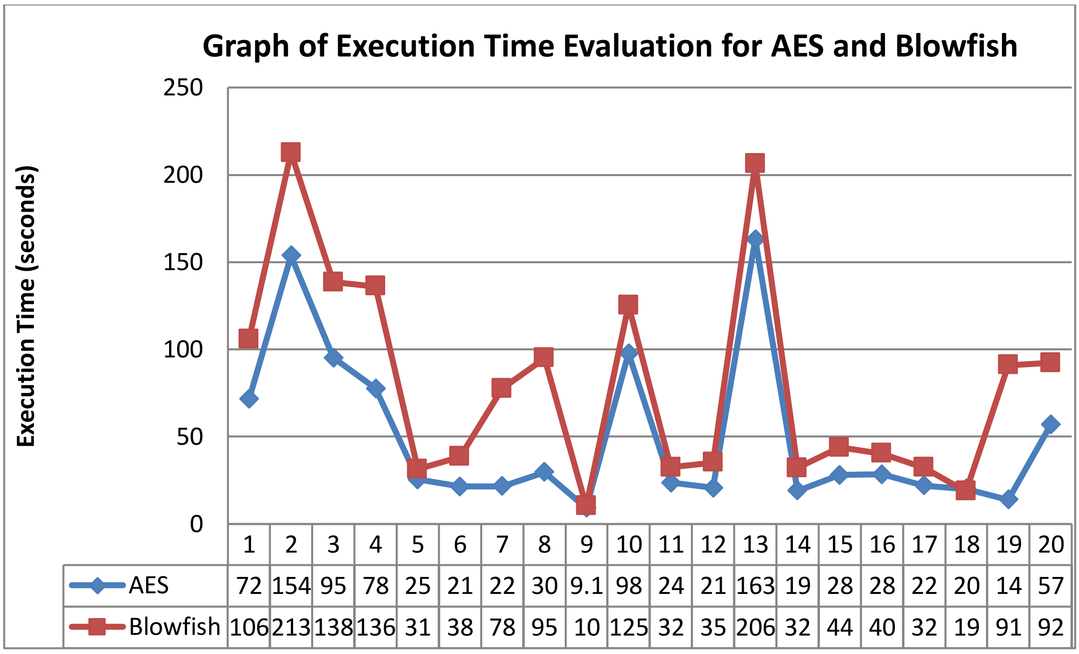


Figure 3:Bar graph showing the comparison of AES and Blowfish algorithms

Figure 3 compares AES and Blowfish across four performance metrics: encryption time, execution time, throughput, and MSE. AES shows lower encryption and execution times than Blowfish, indicating faster processing. In throughput, AES slightly outperforms Blowfish, demonstrating better data-handling efficiency. The most significant difference appears in MSE, where Blowfish records a much higher value, suggesting greater inaccuracy or data distortion during encryption and decryption. Overall, the graph indicates that AES performs more efficiently and reliably across all metrics, making it the superior algorithm in terms of speed, throughput, and accuracy.



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AES | 72 | 154 | 95 | 78 | 25 | 21 | 22 | 30 | 9.1 | 98 | 24 | 21 | 163 | 19 | 28 | 28 | 22 | 20 | 14 | 57 |
| Blowfish | 106 | 213 | 138 | 136 | 31 | 38 | 78 | 95 | 10 | 125 | 32 | 35 | 206 | 32 | 44 | 40 | 32 | 19 | 91 | 92 |

Figure 4:Line graph showing the comparison of AES and Blowfish algorithms

Figure 4 shows the comparison of the execution times of AES and Blowfish across 20 test samples. Overall, AES consistently performs faster, showing lower execution times in most samples. Blowfish displays higher peaks, particularly at samples 2, 12, and 19, indicating slower and more unstable performance. Both algorithms share a similar pattern in fluctuations, but Blowfish's values remain generally higher. AES demonstrates greater stability and efficiency, maintaining lower execution times throughout the evaluation. This suggests that AES is more suitable for applications requiring faster and more predictable execution performance.

## CONCLUSION

The AES algorithm shows moderate execution times across the samples, with values fluctuating between approximately 9 and 163 seconds. This performance reflects AES's design balance between security and speed, where its efficient key scheduling and encryption rounds enable relatively fast processing compared to other algorithms. Despite some fluctuations that AES is still a popular and reliable encryption option, regardless of input variations or system conditions, utilized in a variety of applications because of its dependability and steady pace. Blowfish consistently records the highest execution times among the algorithms tested, with values ranging from around 10 to over 212 seconds. This elevated processing time is attributable to Blowfish's more complex key expansion and encryption structure, which imposes higher computational demands. The trade-off for this complexity is generally enhanced security, but it comes at the cost of slower encryption speeds. As such, Blowfish may be less suitable for applications requiring rapid data handling or real-time encryption, especially for large datasets.In conclusion, AES is faster, more efficient, and more accurate, making it better suited for systems demanding high speed, reliability, and strong encryption performance. Blowfish, while functional, lags significantly behind AES in terms of processing speed and accuracy, limiting its suitability for high-performance security environments.

## REFERENCES

1. Abubakar Z. M., Bala M., Mohammed U. & Ahmed. M. K. (2025). Application of Advanced Encryption Standard (AES) for Securing Electronic Banking Transactional Data.DOI: https://doi.org/10.51584/IJRIAS.2025.100800074
2. Al-Maqtari, E. A., & Al-Maqtari, E. A. (2024).Performance evaluation for AES, Blowfish, DES, and 3DES cryptography algorithms.*Partners Universal Innovative Research Publication, 2*(5), 86–95.https://doi.org/10.5281/zenodo.13974870

3. Bello Buhari, A., AfolayanAyodeleObiniyi, A., Kissinger, S., &Sirajo, S. (2019). Performance evaluation of symmetric data encryption algorithms: AES and Blowfish. *Saudi Journal of Engineering and Technology (SJEAT)*, *4*(10), 407–414.https://doi.org/10.36348/SJEAT.2019.v04i10.002
4. Devi, A., Sharma, A., &Rangra, A. (2015). Performance analysis of symmetric key algorithms: DES, AES and Blowfish for image encryption and decryption. *International Journal of Engineering and Computer Science*, *4*(06).
5. Dhamala, N., & Acharya, K. P. (2024). A comparative analysis of DES, AES and Blowfish based DNA cryptography. *Adhyayan Journal*, *11*(11), 69–80.https://doi.org/10.3126/aj.v11i11.67080
6. Ebtihal A. A.&Elham A. A. (2024).PerformanceEvaluation for AES, Blowfish, DES, and 3DES Cryptography Algorithms. Partners Universal Innovative Research Publication (PUIRP), 02(05), 86–95. https://doi.org/10.5281/zenodo.13974870
7. Koukou, Y. M., Othman, S. H., &HerveNkiama, M. M. (2016).Comparative study of AES, Blowfish, CAST-128 and DES encryption algorithm.*IOSR Journal of Engineering*, *6*(6), 1–7.https://doi.org/10.9790/3021-066010107
8. Timur, M. B. B., Royansyah, R., &Kusumaningsih, D. (2025).Comparison of efficiency and security of AES, Blowfish, and ChaCha20 cryptographic algorithms on image and document files.*Innovatics Journal*, *7*(2).
9. Venkatesh V, Swathi L, Tangudu N. and Satishkumar**,** E S. (2025).Implementation and Evaluation of Data Protection in Databases Using Symmetric Encryption Algorithms.J Neonatal Surg [Internet]. 2025, Mar.24 [cited 2025Nov.30];14(8S):440-59. Available from: https://www.jneonatalsurg.com/index.php/jns/article/view/2558DOI: https://doi.org/10.52783/jns.v14.2558