ISSN 2355-0082

Performance Comparison of AES and Blowfish Algorithm

Hendrawan Nur Majid[1], Trihastuti Yuniati[2], Dedy Agung Prabowo[2]
[1,2,3] Informatics Engineering Study Program, Telkom University, Purwokerto Campus, Banyumas, Indonesia
[1]hendrawanx@student.telkomuniversity.ac.id, [2]trihastutiy@telkomuniversity.ac.id,
[3]dedyaprabowo@telkomuniversity.ac.id

Accepted 09 May 2026
Approved 08 June 2026

***Abstract*— The rapid advancement of communication and data storage technologies requires security systems capable of protecting sensitive information from cyber threats. Symmetric cryptography algorithms such as AES and Blowfish are widely used, although they exhibit different performance characteristics. This study compares both algorithms based on processing time, memory usage, and throughput on .txt, .pdf, and .zip files. The testing was conducted through Python implementation using ECB and CTR modes. The results show that AES consistently outperforms Blowfish in terms of speed and throughput. For large files, AES using ECB mode reaches a throughput of 1128.42 MB/s, significantly higher than Blowfish using ECB mode at 92.26 MB/s, with faster encryption time as well, AES using ECB mode within 0.075 seconds while Blowfish using ECB mode within 0.928 seconds. The decryption process shows similar results, where AES using CTR mode achieves 546.70 MB/s, while Blowfish using CTR mode reaches only 81.59 MB/s. Overall, AES is more suitable for fast encryption-decryption on large-scale data processing, whereas Blowfish is more appropriate for small files or low-complexity applications.**

***Index Terms*— AES; Blowfish; cryptography; encryption algorithm; performance comparison**

I. INTRODUCTION

The rapid development of digital communication and data storage has increased the risk of cyber threats and large-scale data breaches, making data security a major concern for organizations and governments [1], [2]. Cryptography plays a crucial role in protecting sensitive information by ensuring data confidentiality and preventing unauthorized access [3], [4]. Among cryptographic approaches, symmetric encryption is widely used due to its efficiency, with Advanced Encryption Standard (AES) and Blowfish being two prominent algorithms [5].

AES and Blowfish have different design structures and performance characteristics. AES is a modern encryption standard known for its high security and efficiency, whereas Blowfish offers flexibility through its variable key length and simple structure [6], [7], [8]. Previous studies comparing symmetric encryption algorithms such as AES, 3DES, and Blowfish show that Blowfish may achieve better performance and throughput, while AES remains superior in terms of security against various threats [9]. Although previous studies suggest Blowfish outperforms AES and 3DES, further research is necessary because performance results vary depending on implementation context, system architecture, and data size. Additionally, earlier works often neglected the impact of modes of operation, which can significantly influence both performance and security, making a more comprehensive evaluation essential.

Therefore, this study focuses on a comparative performance analysis of AES and Blowfish based on encryption and decryption time, memory usage, and throughput on Electronic Codebook (ECB) and Counter Mode (CTR) mode to determine the most suitable algorithm for modern information security systems. ECB is still included in this research as a baseline because it provides maximum performance with minimal computational overhead, allowing direct evaluation of the raw efficiency of encryption algorithms [10], [11]. However, its known security weaknesses highlight the importance of comparing it with more secure modes like CTR to understand the trade-off between performance and security.

The urgency of this research lies in the increasing demand for efficient and secure encryption mechanisms that can adapt to different system requirements. Modern computing environments, including cloud computing, mobile systems, and Internet of Things (IoT) devices, require encryption algorithms that not only provide strong security but also maintain high performance in terms of processing speed and resource consumption [8], [12]. Choosing an inappropriate algorithm may lead to inefficiency, excessive resource usage, or even potential security risks [13]. Therefore, it is essential to identify which algorithm performs better under certain conditions.

A comparative approach is used in this research because it provides a systematic method to evaluate and analyse the strengths and weaknesses of different algorithms under the same conditions. By directly comparing AES and Blowfish using identical datasets,

hardware environments, and evaluation parameters, the research can produce objective and measurable results. This approach allows researchers to identify performance differences in terms of encryption and decryption time; measure efficiency based on throughput and resource utilization; and evaluate scalability when handling different data sizes.

Moreover, a comparative study is important because theoretical advantages of algorithms do not always reflect their real-world performance. For instance, an algorithm that appears efficient in theory may perform differently when implemented in a specific programming environment or hardware architecture. Through empirical testing, this research aims to bridge the gap between theoretical design and practical implementation.

In addition, the comparison between AES and Blowfish is particularly relevant because they represent two different cryptographic design approaches: AES with its substitution-permutation structure [14] and Blowfish with its Feistel network [15]. Analysing these differences contributes not only to performance evaluation but also to a better understanding of how algorithm design impacts efficiency and security.

Therefore, this research is expected to provide valuable insights into the performance and suitability of AES and Blowfish, helping developers, researchers, and organizations make informed decisions when selecting encryption algorithms for their systems.

II. Literature Review

Several previous studies have evaluated the performance of various encryption algorithms in different application contexts. The study in [16] compares 3DES, AES, Blowfish, and RSA in cloud data storage and classification, demonstrating that Blowfish achieves the fastest encryption performance, while RSA exhibits the highest computational cost in both time and memory usage. Similarly, [17] highlights the efficiency of AES compared to RSA, showing that AES maintains significantly faster and more stable encryption times across varying data sizes, whereas RSA performance increases linearly with input size. These findings suggest that symmetric algorithms are generally more suitable for applications requiring high speed and efficiency.

Further research has explored the comparative performance of other symmetric algorithms. The study in [18] compares Blowfish and Twofish in terms of time and space complexity, showing that Blowfish excels in processing speed, while Twofish offers better memory efficiency. These results indicate that algorithm performance is often influenced by trade-offs between speed and resource consumption.

Another study [19] evaluates multiple AES variants (Rijndael) with Serpent, and Twofish on Android devices, demonstrating that Rijndael provides the best overall performance in terms of encryption speed and CPU usage. Meanwhile, Serpent and Twofish show advantages in memory efficiency and encryption strength, as measured by the avalanche effect. This highlights that the selection of cryptographic algorithms should depend on specific application requirements, including performance, memory efficiency, and security level.

Despite these extensive studies, existing literature still lacks comprehensive evaluation of AES and Blowfish in the context of modern cryptographic challenges, particularly when considering different modes of operation and resistance to evolving attack techniques. Therefore, this research aims to further investigate the performance and resilience of AES and Blowfish under contemporary conditions, providing a more complete understanding of their effectiveness in modern cryptographic applications.

III. Research Methodology

This section presents the research methodology employed in this study, covering the experimental workflow, hardware and software specifications, test data setup, implementation of encryption and decryption processes, performance evaluation metrics, and data analysis techniques to ensure the validity and reliability of the findings. The comparison is conducted by measuring encryption time, decryption time, memory usage, and throughput under controlled experimental conditions, which are commonly used performance metrics in cryptographic evaluation [21].

A. Experimental Workflow

AES and Blowfish are implemented using a uniform key length of 256 bits to ensure an equivalent security level during evaluation, as adopted in several prior studies on file encryption [9]. Both algorithms are tested using two block cipher modes of operation, namely ECB and CTR, which are commonly applied in symmetric cryptographic systems [22], [23]. The experimental approach is used to measure and evaluate the performance of both algorithms based on several parameters, including encryption time, decryption time, memory usage, and throughput.

The experiments are conducted by implementing AES and Blowfish algorithms in a graphical user interface (GUI) based application. This application is designed to perform encryption and decryption processes while automatically recording performance metrics. The results obtained from these experiments are then analysed to identify performance differences between the two algorithms under various testing conditions. The steps for this research can be seen in Figure 1.

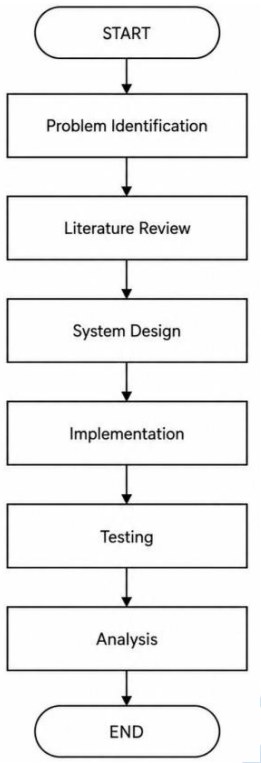

Fig. 1. Experimental Workflow

The research method of this study consists of several stages, which are as follows:

- **Problem Identification:** Identification of the problems that occur is carried out.
- **Literature Review:** A literature review is conducted to understand the concepts of cryptography, symmetric encryption, and the characteristics of AES and Blowfish algorithms.
- **System Design:** A GUI based system is designed to facilitate file selection, encryption, decryption, and performance measurement.
- **Implementation:** AES and Blowfish algorithms are implemented in two operating modes, ECB and CTR, using the Python programming language.
- **Testing:** Encryption and decryption tests are performed on different types and sizes of files to collect performance data including running time, memory usage, and throughput.
- **Analysis:** The collected data are analyzed to compare the performance of AES and Blowfish based on predefined parameters.
-

B. Hardware and Software Specification

To ensure consistency and reliability of the experimental results, this research was conducted using a specific hardware and software configuration. The specifications are summarized in the Table 1 and Table 2.

TABLE I. HARDWARE SPECIFICATION

Component	Specification Description
Processor (CPU)	Intel Core i5
Memory (RAM)	16 GB
Storage	1 GB SSD
Operating Device	Laptop

The hardware used in this research consists of a computer system equipped with an Intel Core i5 processor, sufficient RAM capacity, and solid-state storage to support efficient encryption and decryption processes.

TABLE II. SOFTWARE SPECIFICATION

Software Component	Specification Description
Operating System	Windows 10 (64-bit)
Programming Language	Python
Development Tools	Visual Studio Code
Cryptography Library	PyCryptodome
Supporting Software	Microsoft Excel for analysis

The system was implemented using Python programming language, supported by cryptographic libraries and development tools to facilitate encryption, decryption, and performance measurement.

C. Test Data Setup

The test data used in this study consist of digital files with common formats, namely **.txt*, **.pdf*, and **.zip*. These file types are selected because they represent different data characteristics. File sizes are classified into three categories [19], [21]:

- Small files: ≤ 100 KB
- Medium files: 100 KB – 5 MB
- Large files: > 5 MB

This classification aims to observe the impact of file size variations on the performance of AES and Blowfish algorithms.

D. Encryption and Decryption Process

The encryption and decryption processes are conducted using AES and Blowfish algorithms with ECB and CTR modes of operation. Both algorithms use a 256-bit key to ensure an equal level of security during performance comparison. For CTR mode, a counter value combined with a nonce is used to generate a keystream, while ECB mode encrypts each plaintext block independently using the same key.

To validate functional correctness, the decrypted output is compared with the original file to ensure data integrity, as applied in prior cryptographic implementation studies [24]. Each file is encrypted and subsequently decrypted using the same secret key. During these processes, the system records encryption time, decryption time, memory usage, and throughput.

E. Performance Evaluation Parameters

The performance of AES and Blowfish is evaluated based on the following parameters [9]:

- **Encryption Time**: The time required to encrypt a file, measured in seconds.
- **Decryption Time**: The time required to decrypt an encrypted file back to its original form.
- **Memory Usage**: The amount of memory consumed during encryption and decryption processes.
- **Throughput**: The rate at which data is processed during encryption or decryption, measured in MB/s.

F. Data Analysis Technique

The collected experimental data are analysed using comparative analysis techniques. Performance results of AES and Blowfish are compared across different file types, file sizes, and modes of operation (ECB and CTR). The analysis focuses on identifying performance trends, efficiency differences, and suitability of each algorithm for practical data security applications.

IV. RESULT AND ANALYSIS

This chapter presents experimental results and provides a comprehensive discussion of the performance evaluation of the AES and Blowfish.

B. System Design

The system workflow is designed to provide a complete benchmarking cycle, starting from data input, encryption–decryption processing, performance measurement, and ending with flexible output handling and result visualization. This design ensures both usability and comprehensive evaluation of the encryption algorithms. The system workflow can be seen in Figure 2.

The system begins when the user selects a plaintext file as input. After selecting the file, the user initiates the process by clicking the “Run Benchmark” button. The system then performs the encryption process using the selected algorithm and mode of operation, followed by the decryption process to validate correctness.

During this process, the system measures key performance metrics such as encryption time, decryption time, and throughput. Once the process is completed, the output results are displayed to the user.

After viewing the results, the user is provided with several options, including saving the output to an Excel file, displaying performance graphs, saving the resulting file (either ciphertext or plaintext), performing an additional decryption process, or resetting all inputs. If the user chooses to decrypt a file, the system executes the decryption process and displays the resulting plaintext. The workflow ends after all selected operations are completed.

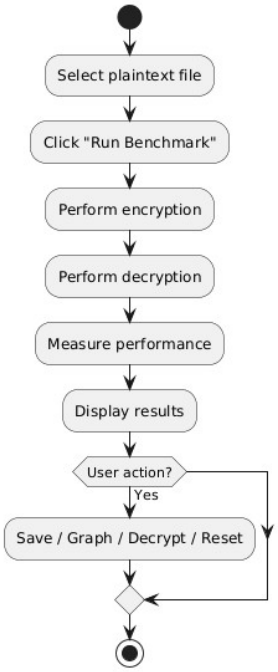

Fig. 2. System Workflow of Encryption and Decryption Benchmark Process

Implementation

The AES and Blowfish algorithms are implemented using the PyCryptodome library in the Python programming language. Figure 3 shows the results of the program implementation.

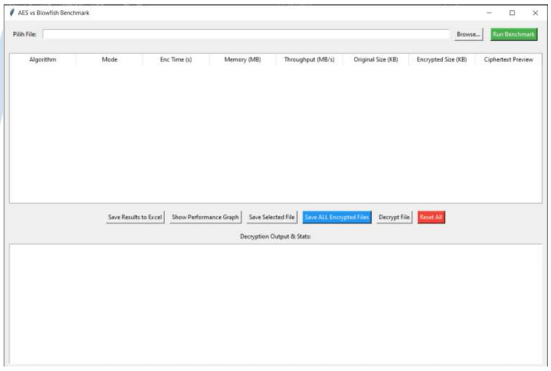

Fig. 3. Program implementation result

Testing and Analysis

This chapter presents the results of performance testing. The evaluation was performed using **.txt*, **.pdf*, and **.zip*, which represent small, medium, and large file sizes.

1) Encryption Performance Analysis

The experimental findings reveal that AES consistently outperforms Blowfish in encryption time

and throughput across all evaluated file types, indicating its superior efficiency in handling diverse data formats.

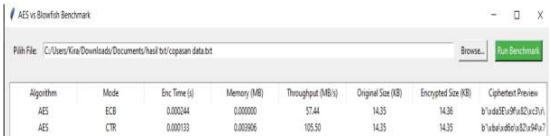

Fig. 4. Testing result of encryption using AES on *.txt* file

The test results of the AES algorithm on a *.txt file with an original size of 14.35 KB using ECB and CTR are presented in Figure 4. The results indicate that the AES-CTR mode demonstrates significantly superior performance in terms of speed. Specifically, AES-CTR achieved a throughput of 105.50 MB/s, which is nearly twice as fast as AES-ECB, which only reached 57.44 MB/s. This performance advantage is also reflected in the encryption time, where AES-CTR requires only 0.000133 seconds, compared to AES-ECB, which takes 0.000244 seconds.

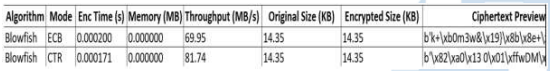

Fig. 5. Testing result of encryption using Blowfish on *.txt*

On the other hand, using the same text file, the results indicate that the Blowfish-CTR mode demonstrates superior performance compared to Blowfish-ECB. Specifically, Blowfish-CTR achieves a throughput of 81.74 MB/s with an encryption time of only 0.000171 seconds. In contrast, Blowfish-ECB exhibits lower performance, with a throughput of 69.95 MB/s and a slightly longer encryption time of 0.000230 seconds. In this experiment, both ECB and CTR modes of Blowfish recorded negligible memory usage, measured at approximately 0.000000 MB, indicating minimal memory overhead during the encryption process.

Tables III–V summarize the encryption performance comparison between AES and Blowfish for different file types, where Table III corresponds to *.txt files, Table IV to *.pdf files, and Table V to *.zip files.

TABLE III. ENCRYPTION PERFORMANCE ON TXT FILES

Algorithm	Mode	Enc. Time (s)	Memory (MB)	Throughput (MB/s)
AES	ECB	0.000244	0.000000	57.44
AES	CTR	0.000133	0.003906	105.50
Blowfish	ECB	0.000200	0.000000	69.65
Blowfish	CTR	0.000171	0.000000	81.74

TABLE IV. ENCRYPTION PERFORMANCE ON PDF FILE

Algorithm	Mode	Enc. Time (s)	Memory (MB)	Throughput (MB/s)
AES	ECB	0.001276	1.511719	1179.27
AES	CTR	0.002264	1.507812	664.43
Blowfish	ECB	0.017928	1.507812	83.91
Blowfish	CTR	0.018114	1.507812	83.05

For *.pdf files, AES consistently demonstrates superior performance compared to Blowfish. AES-ECB achieves the fastest encryption time of 0.001276 seconds with a throughput of 1179.27 MB/s, followed by AES-CTR at 0.002264 seconds and 664.43 MB/s. In contrast, Blowfish requires considerably longer processing times, resulting in significantly lower throughput, confirming AES's efficiency advantage for larger and more complex file types.

TABLE V. ENCRYPTION PERFORMANCE ON ZIP FILE

Algorithm	Mode	Enc. Time (s)	Memory (MB)	Throughput (MB/s)
AES	ECB	0.075952	85.707031	1128.42
AES	CTR	0.112998	85.707031	758.47
Blowfish	ECB	0.928997	85.707031	92.26
Blowfish	CTR	0.976330	85.769531	87.78

For *.zip, the performance gap becomes more pronounced. AES-ECB achieves an encryption time of 0.075952 seconds with a throughput of 1128.42 MB/s, whereas AES-CTR records 0.112998 seconds with 758.47 MB/s. Blowfish, on the other hand, requires significantly longer encryption times of 0.928097 seconds (ECB) and 0.976330 seconds (CTR), with throughput values limited to 92.26 MB/s and 87.78 MB/s, respectively.

For small sized files such as *.*txt*, both algorithms exhibit similar performance; however, AES in CTR mode achieves the highest throughput. In medium-sized files such as *.*pdf*, AES demonstrates significantly faster encryption times and higher throughput compared to Blowfish. This performance gap becomes more pronounced when processing large files (*.*zip*), where AES shows superior efficiency, especially in ECB mode. These results indicate that AES scales more effectively when handling large datasets, making it more suitable for applications involving high-volume data encryption.

2) Decryption Performance Analysis

The decryption performance results show a similar trend to the encryption results. AES, particularly in CTR mode, records shorter decryption times and higher throughput compared to Blowfish across all file types. For small text files, the decryption time difference between the two algorithms is minimal. However, for medium and large files, AES

demonstrates a clear advantage in terms of processing speed and throughput. Tables IV-VI present the decryption performance comparison between AES and Blowfish for *.txt, *.pdf, and *.zip files.

TABLE VI. DECRYPTION PERFORMANCE ON TXT FILE

Algorithm	Mode	Dec Time (s)	Memory (MB)	Throughput (MB/s)
AES	ECB	0.004384	0.023438	3.20
AES	CTR	0.006108	0.355469	2.29
Blowfish	ECB	0.000390	0.000000	35.90
Blowfish	CTR	0.001122	0.015625	12.49

For *.txt files, AES-CTR completes decryption in 0.0006 seconds, while Blowfish-CTR requires 0.0011 seconds. Although both algorithms perform quickly for small files, AES remains faster. *D.*

TABLE VII. DECRYPTION PERFORMANCE ON PDF FILE

Algorithm	Mode	Dec. Time (s)	Memory (MB)	Throughput (MB/s)
AES	ECB	0.007883	1.507812	190.83
AES	CTR	0.002411	1.507812	623.92
Blowfish	ECB	0.024608	1.507812	61.13
Blowfish	CTR	0.029752	1.507812	50.56

In *.pdf files, AES-CTR records a decryption time of 0.0024 seconds with a throughput of 623.92 MB/s, whereas Blowfish-CTR requires 0.0029 seconds with a throughput of 50.56 MB/s.

TABLE VIII. DECRYPTION PERFORMANCE ON ZIP FILE

Algorithm	Mode	Dec. Time (s)	Memory (MB)	Throughput (MB/s)
AES	ECB	0.173287	85.707031	494.59
AES	CTR	0.156770	85.707031	546.70
Blowfish	ECB	0.987407	85.707031	86.80
Blowfish	CTR	1.050477	85.769531	81.59

For *.*zip* files, AES-CTR completes decryption in 0.15 seconds with a throughput of approximately 546–546.70 MB/s, while Blowfish-CTR requires 1.05 seconds with a throughput of only 81.59 MB/s.

Additionally, the file sizes after decryption are identical to the original file sizes across all tests, confirming that the decryption process successfully restores the original data without any loss or modification.

3) Memory Usage Analysis

The memory usage analysis shows that both AES and Blowfish exhibit relatively similar memory consumption during the encryption and decryption processes. The measured memory usage ranges from approximately 0.000000 MB to 85.769531 MB, depending on the type and size of the input file processed. This variation indicates that memory usage is primarily influenced by file characteristics rather than the choice of encryption algorithm.

Furthermore, the results suggest that neither AES nor Blowfish introduces significant additional memory overhead beyond the baseline requirements of the input data. The similarity in memory consumption across both algorithms implies that memory efficiency is not a distinguishing factor in their overall performance. Instead, the observed performance differences, particularly in encryption time and throughput, are more strongly attributed to differences in algorithm structure and computational efficiency rather than memory utilization.

Discussion

The combined visualization (Figure 6) illustrates the overall performance comparison between AES and Blowfish across different file types (TXT, PDF, and ZIP) and processes (encryption and decryption) in terms of execution time, throughput, and memory usage.

The results show that AES consistently exhibits superior performance, particularly for medium and large file sizes, where it achieves significantly lower execution times and substantially higher throughput compared to Blowfish. The higher throughput achieved by AES indicates its ability to process data more efficiently within a shorter time period, making it more suitable for applications that require high-speed data processing and real-time security. AES performs better than Blowfish in throughput and execution time due to its optimized substitution–permutation structure and hardware acceleration support, which enable more efficient parallel processing compared to the more complex key scheduling and Feistel structure of Blowfish. This performance gap becomes more pronounced as the file size increases, indicating that AES has better scalability and computational efficiency.

In contrast, Blowfish demonstrates relatively stable but slower performance, especially for larger datasets such as ZIP files. Meanwhile, memory usage remains relatively consistent between both algorithms and is primarily influenced by the size of the input file rather than the encryption method. Blowfish may only be competitive in scenarios involving smaller data sizes.

Although this study primarily focuses on performance evaluation, security aspects are inherently considered through the selection of algorithms and modes of operation. AES is widely recognized as a secure standard encryption algorithm, while Blowfish, although efficient, has known limitations such as its

smaller block size. Furthermore, the inclusion of ECB and CTR modes provides an implicit security comparison, as ECB represents a less secure baseline, whereas CTR offers stronger protection against pattern leakage and supports secure implementation in modern systems.

To address this limitation, this study emphasizes that performance and security are closely interconnected, and the choice of encryption technique must account for both factors simultaneously. The results highlight that while some configurations may achieve higher speed (e.g., ECB), they do so at the expense of security, whereas modes such as CTR provide a better balance between efficiency and security. Therefore, the findings of this research not only contribute to performance benchmarking but also provide insight into the practical security implications of selecting specific algorithms and modes of operation.

Overall, the findings suggest that AES is more appropriate for modern information systems that demand both high security and high computational efficiency, such as cloud storage, secure communication systems, and large-scale data processing platforms. Blowfish remains a viable alternative for applications involving small file sizes or environments with lower computational complexity requirements.

V. CONCLUSIONS

This study demonstrates that the evaluation of cryptographic algorithms must consider both algorithm design and mode of operation, as performance outcomes are strongly influenced by their interaction. While AES consistently outperforms Blowfish in terms of encryption speed, decryption time, and throughput, the findings also indicate that performance alone is not sufficient for determining algorithm suitability. Instead, an appropriate balance between efficiency, scalability, and security must be considered, particularly when selecting modes such as ECB and CTR that introduce different trade-offs.

The results further highlight that memory usage is relatively similar between AES and Blowfish, suggesting that computational efficiency (rather than resource consumption) is the primary factor distinguishing their performance. In this context, AES combined with more secure and efficient modes (e.g., CTR) provides a more practical solution for modern applications that require both high-speed processing and robust security. Conversely, although Blowfish shows stable performance, its advantages are more limited to specific scenarios with less demanding performance requirements.

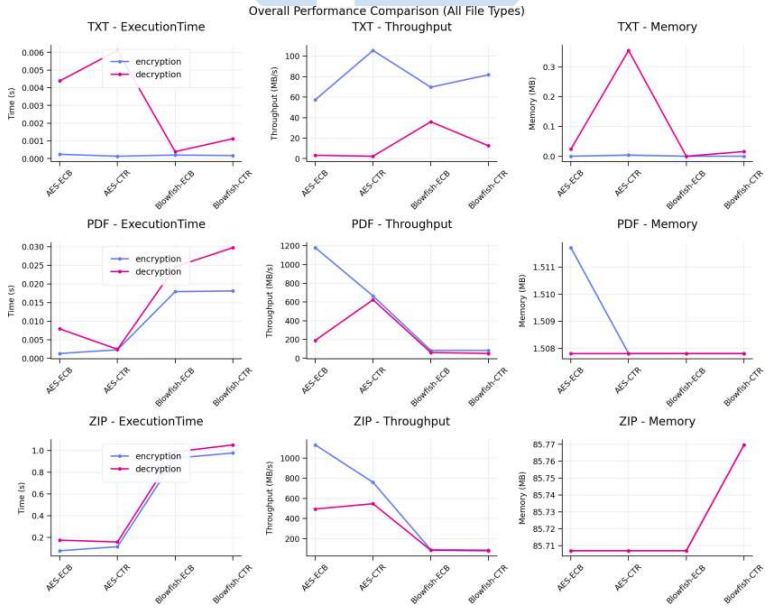

Fig. 6. Overall performance comparison of AES and Blowfish across different file types (TXT, PDF, and ZIP) for both encryption and decryption processes

From a practical standpoint, this research provides guidance for system designers and developers in selecting encryption strategies tailored to application needs, such as high-throughput systems, real-time communication, or resource-constrained environments. Academically, this study contributes by emphasizing the importance of evaluating cryptographic algorithms together with their modes of operation, offering a more realistic representation of

real-world implementations rather than isolated algorithm comparisons.

Finally, this study underscores the need for further research that incorporates more diverse environments, larger datasets, and emerging security challenges, including resistance to modern cryptographic attacks. Such future work is essential to ensure that performance evaluations remain relevant in the context of continuously evolving computing technologies.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Telkom University for providing the facilities and academic support necessary to conduct this research. The supportive research environment and available resources have significantly contributed to the successful completion of this study. We also deeply appreciate the institution's commitment to advancing scientific research and innovation.

REFERENCES

[1] A. Sykot, M. S. Azad, W. R. Tanha, B. M. M. Morshed, S. E. U. Shubha, and M. R. C. Mahdy, "Multi-layered security system: Integrating quantum key distribution with classical cryptography to enhance steganographic security," *Alexandria Eng. J.*, vol. 121, pp. 167–182, 2025, doi: https://doi.org/10.1016/j.aej.2025.02.056.

[2] L. Judijanto, P. D. Persadha, I. Susilowati, H. K. Reza, and M. Susanti, "Analisis Keamanan Data dan Perlindungan Privasi dalam Pengelolaan Big Data: Tinjauan Teknologi Enkripsi dan Anonimisasi," *J. Penelit. Inov.*, vol. 5, no. 2, pp. 1991–2000, May 2025, doi: 10.54082/jupin.1151.

[3] A. Desianty and I. Imelda, "Systematic Literature Review: Cybersecurity by Utilizing Cryptography Using the Data Encryption Standard (DES) Algorithm," *J. Tek. Inform.*, vol. 17, no. 1, pp. 30–39, May 2024, doi: 10.15408/jti.v17i1.37256.

[4] T. R. Syafutra, K. Khairil, and E. Suryana, "The Implementation Of Modern Cryptography On Document Data Security," *J. Media Comput. Sci.*, vol. 1, no. 2, Jul. 2022, doi: 10.37676/jmcs.v1i2.2742.

[5] R. Indrayani, P. Ferdiansyah, and M. Koprawi, "Analisis Penggunaan Kriptografi Metode AES 256 Bit pada Pengamanan File dengan Berbagai Format," *Digit. Transform. Technol.*, vol. 4, no. 2, pp. 1245–1251, Feb. 2025, doi: 10.47709/digitech.v4i2.5457.

[6] A. R. Y. Simatupang, D. Annura, Y. S. Dachi, and D. R. Harries, "Keamanan Kriptosistem Modern Berdasarkan Algoritma Kriptografi Kunci Publik," *J. Siteba*, vol. 2, no. 1, pp. 1–6, 2023.

[7] Sitinjak, N. Marsan, R. O. Batubara, and F. Ikorasaki, "Perancangan dan Implementasi Algoritma Blowfish Untuk Keamanan Data File Citra Digital," *J. Widya*, vol. 5, no. 1, pp. 486–481, 2024.

[8] R. Ganesh, B. U. I. Khan, A. R. Khan, A. Bin Kamsin, M. K. Yang, and J. S. Jeong, *A panoramic survey of the advanced encryption standard: from architecture to security analysis, key management, real-world applications, and post-quantum challenges*, vol. 15, no. 5. 2025. doi: 10.1007/s10207-025-01116-x.

[9] 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. 4, pp. 6473–6486, 2025, doi: 10.35444/IJANA.2024.16404.

[10] D. Sarangi, "A Comparative Study of AES Encryption Modes and Hashing for Blockchain Applications," 2025. [Online]. Available: https://norma.ncirl.ie/id/eprint/7729

[11] "Block Cipher Modes of Operation Explained (ECB, CBC, CTR)," 2026. [Online]. Available: https://www.codinglad.com/blogs/block-cipher-modes-of-operation

[12] M. K. Yang and J. S. Jeong, "Optimized Hybrid Central Processing Unit–Graphics Processing Unit Workflow for Accelerating Advanced Encryption Standard Encryption: Performance Evaluation and Computational Modeling," *Appl. Sci.*, vol. 15, no. 7, 2025, doi: 10.3390/app15073863.

[13] K. Assa-Agyei and F. Olajide, "A Comparative Study of Twofish, Blowfish, and Advanced Encryption Standard for Secured Data Transmission," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 3, pp. 393–398, 2023, doi: 10.14569/IJACSA.2023.0140344.

[14] B. Sarkar, A. Saha, D. Dutta, G. De Sarkar, and K. Karmakar, "A Survey on the Advanced Encryption Standard (AES): A Pillar of Modern Cryptography," *Int. J. Comput. Sci. Mob. Comput.*, vol. 13, no. 4, pp. 68–87, 2024, doi: 10.47760/ijcsmc.2024.v13i04.008.

[15] S. SR, U. N, C. R, and A. CM, "Comparison Between Encryption Algorithms: A Performance and Security Perspective," *Int. J. Sci. Technol.*, vol. 16, no. 3, pp. 1–7, 2025, doi: 10.71097/ijsat.v16.i3.7986.

[16] D. Commey, S. Griffith, and J. Dzisi, "Performance comparison of 3DES, AES, Blowfish and RSA for Dataset Classification and Encryption in Cloud Data Storage," *Int. J. Comput. Appl.*, vol. 177, no. 40, pp. 17–22, Feb. 2020, doi: 10.5120/ijca2020919897.

[17] D. B. Nawawi, M. M. Huda, and T. Prabowo, "Perbandingan Enkripsi Advanced Encryption Standard dan Enkripsi Rivest Shamir Adleman," *G-Tech J. Teknol. Terap.*, vol. 8, no. 3, pp. 1649–1655, Jul. 2024, doi: 10.33379/gtech.v8i3.4452.

[18] C. Kurniawan, M. F. Magfur, and F. Fauziah, "Analisis Perbandingan Ruang dan Waktu Algoritma Enkripsi Blowfish dan Twofish Pada Enkripsi dan Dekripsi Berkas Menggunakan Modul Python," *E-Link J. Tek. Elektro dan Inform.*, vol. 19, no. 1, p. 7, May 2024, doi: 10.30587/e-link.v19i1.6592.

[19] M. B. P. Sansaya and A. Farisi, "Perbandingan Kinerja Algoritma Kandidat AES Dalam Enkripsi dan Dekripsi File Dokumen," *MDP Student Conf.*, vol. 2, no. 1, pp. 282–289, Apr. 2023, doi: 10.35957/mdp-sc.v2i1.4367.

[20] E. Ozer and H. Aydos, "Performance and Security of AES, DES, and RSA in Hybrid Systems: An Empirical Analysis of Triple Encryption," *Int. J. Comput. Exp. Sci. Eng.*, vol. 10, no. 4, Dec. 2024, doi: 10.22399/ijcesen.694.

[21] R. Fadlan, F. Siregar, N. Dly, S. Wulandari, N. A. Siregar, and I. Rusydi, "Comparative Analysis ff Encryption and Decryption Speed of AES and Blowfish Algorithms," *Interdiscip. J. Glob. Multidiscip.*, vol. 2, no. 1, pp. 449–458, 2026.

[22] R. E. Hiromoto, A. Carlson, and M. Singh, "Breaking the Counter (CTR) Mode," in *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, Jan. 2025, pp. 00183–00188. doi: 10.1109/CCWC62904.2025.10903932.

[23] I. A. W. Arnawa, P. E. W. H. C. A. A., and G. B. Putra, "Perbandingan Waktu Enkripsi antara Metode Electronic Code Book (ECB) dan Chipher Block Chaining (CBC) dalam Algoritma Blowfish," *J. Ilmu Komput. Indones.*, vol. 5, no. 1, pp. 50–54, 2020, doi: https://doi.org/10.23887/jik.v5i1.3056.

[24] N. I. Manik and H. J. Fernando, "Program Aplikasi Algoritma Blowfish Pada Sistem Keamanan Data File," *J. Inf. Syst. Applied, Manag. Account. Res.*, vol. 7, no. 1, p. 10, Feb. 2023, doi: 10.52362/jisamar.v7i1.984.