The Open Biomedical Engineering Journal
ISSN: 1874-1207
DOI: 10.2174/0118741207392946260209171217, 2026, 20, e18741207392946

RESEARCH ARTICLE
OPEN ACCESS

Analyzing the Security and Performance Trade-offs of Common Symmetric Encryption Algorithms in Communication Systems

Kafayat Odunayo Tajudeen[1,2], Ahmed Oloduowo Ameen[2], Abidemi Emmanuel Adeniyi[2,3,4,*], Olu Randle[5] and Oluwasegun Julius Aroba[6,7]

[1] *Department of Computer Science, Al-Hikmah University, Ilorin, Nigeria*
[2] *Department of Computer Science, University of Ilorin, Ilorin, Nigeria*
[3] *Department of Computer Science, Bowen University, Iwo, Nigeria*
[4] *Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India*
[5] *University of the Witwatersrand, Computer Science, Johannesburg, South Africa*
[6] *Department of Operations and Quality Management, Durban University of Technology, Durban, South Africa*
[7] *Centre for Ecological Intelligence, Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa*

Abstract:
The primary benefit of symmetric key encryption is that it requires less computing power than asymmetric encryption. Additionally, as every symmetric encryption method has advantages and disadvantages, it is vital to assess how well popular symmetric encryption algorithms perform in order to determine their suitability for a range of efficiency scenarios and applications. The expansion of communication applications has made security a crucial concern for file storage and transfer. In computer networks, encryption, which is based on the science of cryptography, is necessary to safeguard data transmission. The two kinds of cryptography are symmetric and asymmetric encryption. Because of its resource-constrained characteristics, asymmetric encryption has the drawback of computational complexity, which renders it unsuitable for communication applications. Using the Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Blowfish, this study evaluated the security of plaintext files using three secret key cryptography techniques. The simulation application was developed using the Python programming language. Text files of 0.5 MB, 1 MB, 2 MB, 5 MB, 10 MB, and 20 MB were used to evaluate the avalanche effect, execution time, and throughput. The encryption algorithms employed 14 rounds, 16 rounds, and 16 rounds for AES, DES, and Blowfish, respectively. Conclusively, AES outperformed DEB and Blowfish in security performance, but Blowfish outperformed them in both time and throughput.

Keywords: Communication applications, Symmetric encryption, AES, DES, Blowfish.

Received: March 12, 2025
Revised: May 27, 2025
Accepted: October 07, 2025
Published: February 18, 2026

*Address correspondence to this author at the Department of Computer Science, University of Ilorin, Ilorin, Nigeria; Department of Computer Science, Bowen University, Iwo, Nigeria and Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India; E-mail: abidemi.adeniyi@bowen.edu.ng

Cite as: Tajudeen K, Ameen A, Adeniyi A, Randle O, Aroba O. Analyzing the Security and Performance Trade-offs of Common Symmetric Encryption Algorithms in Communication Systems. Open Biomed Eng J, 2026; 20: e18741207392946. http://dx.doi.org/10.2174/0118741207392946260209171217

1. INTRODUCTION

Applications are becoming increasingly important to people and companies in the digital age for data processing, storage, and communication [1]. The security of sensitive information communicated *via* intrinsically unreliable media is a major problem because of this dependency [2]. For example, social media sites and messaging apps are vulnerable to cyber attacks that target private communications and login information [3, 4].

Major security concerns are raised by the dependency on digital communications and storage systems, especially with regard to maintaining the security of private communications *via* computer networks. Symmetric and asymmetric encryption are the building blocks of encryption, which play a vital role in maintaining the security of digital communications [5]. Asymmetric encryption is less useful due to slower speed and increased computational requirements in comparison to symmetric encryption. Asymmetric encryption may be useful only for a limited amount of data transfer due to these reasons [6, 7]. Therefore, due to these limitations, symmetric encryption methods such as AES, DES, and Blowfish are preferred because of their increased speed and lower computational requirements. However, using these symmetric encryption methods poses major research problems with regard to determining the best encryption method [8, 9]. Asymmetric encryption is highly secure; however, due to its increased computational requirements, it may be less useful in real-time applications such as cloud-based communications systems, mobile devices, and Internet of Things communications systems [10]. Therefore, due to this constraint, symmetric encryption methods are preferred. However, these symmetric encryption methods are less successful [11]. Their cryptanalytic attack resistance, scalability, memory usage, and computing power are significantly different. Determining the best symmetric encryption method with regard to security-performance trade-off in a specific scenario is a major research problem [12-14]. For example, even though algorithms such as AES, DES, and Blowfish are highly popular with regard to securing digital communications, their relative merits and demerits are different based on specific scenarios [15-17]. DES is popular due to its historical significance and lower longevity; Blowfish is popular due to its speed; and AES is popular due to its security features [18].

Yabo *et al*. [19] This paper discusses four of the most frequently adopted models of encryption using VB.NET programming, such as DES, RC2, 3DES, and AES. Ultimately, it measures how well each performs, with a GUI designed to measure times and the size of data. The results show that faster model execution in VB.NET reduces the risk of data breaches. In looking into the future, there is no other way for the sustainable and efficient construction of energy systems in self-powered devices other than exploring new materials and imaginative construction techniques, considering the fast depletion of current traditional energy reserves [20]. Mechanical energy harvesting devices such as TENGs have been attracting increasing interest because of their unique properties. In turn, next-generation bio-inspired technologies depend on stringent handwriting recognition and detection technologies. Among others, a rigorous and highly controlled experimental investigation would point out the trade-offs involved. Finally, in order to enable researchers and developers to select the best encryption algorithm that suits the current communication system requirements, closing this research gap is crucial.

1.1. Motivation

The study has emerged from the growing need to address the increasing complexity of cybersecurity in securely transferring digital data. As online threats become harder to recognize and neutralize, maintaining the confidentiality and integrity of digital information becomes more difficult. In that respect, cryptography is identified as an essential pillar in the areas of application security and protecting data against hostile intrusions. Herein, we attempt to bring forth the best, most secure, and efficient encryption technique by weighing the traditional cryptographic approaches and continuing improvement in the encryption algorithms. This will involve testing older methods with concrete data, then brainstorming to identify where improvements should be made to surface weaknesses. It will be exemplified by exercises in mathematics, including the avalanche effect, throughput of encryption and decryption, and time required for cryptographic analysis. The paper also focuses on AES, DES, and Blowfish after discussing previous research. These cryptograms are in common use within cryptographic practice due to their efficiency regarding block encryption and protection. Three symmetric block cipher techniques are discussed. Some of the factors taken into consideration while choosing source text files for analysis are data type, number of rounds, size of block, key size, and algorithm-specific requirements. A wide range of source text files in different sizes was used while carrying out the testing to investigate how well AES, DES, and Blowfish can perform.

1.2. Contribution

The works presented will contribute in the following ways:

(1) It compares three symmetric encryption algorithms, such as AES, DES, and Blowfish. The performance of these algorithms is evaluated using the avalanche effect, execution time, and throughput; they are implemented using the Python programming language.

(2) Avalanche was tested using different algorithms for the keys of sizes 256-bit, 56-bit, and 32-bit, which showed that AES has the strongest avalanche effect as compared to DES and Blowfish algorithms.

(3) As far as security strength is concerned, AES stands out as the best choice when security is a high priority.

(4) The execution times were calculated with 14 iterations for Blowfish, whereas AES and DES were performed with 16 iterations. This showed that Blowfish is faster than the other two algorithms in processing time. Hence, Blowfish is suitable for use when quick encryption and/or decryption are required.

(5) Performance tests with 256-bit, 56-bit, and 32-bit keys have proven that Blowfish has the highest performance among the three.

(6) Overall, AES and Blowfish have proven themselves to be the best choices for text file encryption in various scenarios, ranging from cloud-based platforms to

resource-constrained devices. - The paper points out the avalanche effect, execution time, and throughput for the algorithms AES, DES, and Blowfish, which demonstrate the balance between security and performance in symmetric cryptography.

2. REVIEW OF RELATED WORK

[21] This paper presents a survey of different cryptographic techniques for data protection, including AES, Blowfish, Twofish, Salsa20, and ChaCha20. In the comparison, ChaCha20 comes out to be the best encryption/decryption technique that provides the best average time, though it shows the lowest throughput in general. The aim of this study is to find the optimal timing and throughput for data encryption and decryption when implemented in Java [22]. One more work analyzes the efficiency of different encryption algorithms, namely AES, Blowfish, DES, and 3DES. Of these, AES is the most efficient and fastest at encrypting data for transmission over a network.

[19] It considers four common algorithms, such as DES, RC2, 3DES, and AES, in VB.NET-based analysis. The performance is measured using a graphical user interface through time and data-size metrics. The results show that AES executes more quickly in VB.NET, which will help decrease the risk of data leakage.

[23] Another investigation studied the performance of AES, CAST, Blowfish, and TE-DES on medical images. The results indicated that E-DES had the least encryption time taken on the tested images: Image A = 4.08s, Image B = 3.93 seconds, Image C = 3.16 seconds, Image D = 3.89 seconds, and Image E = 4.00 seconds. Such a comparison provides further insight into how different image encryption methods would perform.

[10] Three widely used block cipher algorithms, AES, Blowfish, and Twofish, were analyzed for encryption, decryption speeds, and throughput. In terms of encryption and decryption speed, Blowfish performed better than AES. The study employed a Python 3.10 application for data simulation and encryption process, and speed to test each algorithm's efficacy based on performance time and speed.

[24] AES offers the best trade-off between security and speed, according to the study that contrasts the use of neural networks for data encryption with DES, AES, and Blowfish. The study recommends employing picture encryption algorithms and encrypting weight data using encryption techniques to improve data security. The results show that DES and Blowfish together are not as secure as AES and HE, even if they offer somewhat faster encryption and decoding. Future studies should look into picture encryption techniques and advanced privacy-preserving tactics to make digital encryption and decryption activities more secure and private.

[25] The Blowfish, DES, and AES algorithms are discussed in this work along with their benefits, drawbacks, and practical uses. It analyzes their design concepts, encryption algorithms, security analysis, and possible uses in cryptographic environments. The paper offers a thorough grasp of the future directions and possible uses of the Blowfish algorithm in symmetric-key cryptography.

[13] The energy costs of symmetric and asymmetric key systems are thoroughly compared in this work. There are two ways to do this comparison. The first technique employs the Energy Cost Of Data Utilization (ECDU) metric to quantify the worldwide energy expenditures associated with internet data usage. It was discovered that the annual energy used by public-key cryptography applications worldwide is enough to power 1000 UK houses for a whole year. In the second approach, a small-scale network of wireless embedded devices is built using an experimental strategy. In order to assess the computing and communication costs of each solution in a controlled setting, this is then utilized to compare two important establishment strategies, symmetric and asymmetric.

[26] This study, which evaluated the encryption and decryption of document files using the DES and Blowfish algorithms, found that the former uses less memory and makes the file larger while encrypting. The DES algorithm resulted in 3.595% and 3.7975% faster encryption and decryption processing times, respectively. The DES algorithm's file size grew by fewer bytes than that of the Blowfish algorithm. Upon decryption, the file size restored to its initial size. For encryption and decryption, the DES algorithm needed 49.655% and 49.5925% more RAM, respectively.

[27] The study examined AES, 3DES, Blowfish, and Twofish and found that while Blowfish and Twofish have the greatest ciphertext sizes, AES has the fastest encryption and decryption execution times. Additionally, the study discovered that Blowfish and Twofish have the biggest ciphertext sizes, whereas AES and 3DES require less memory for encryption and decoding.

[28] The following paper assesses three prominent encryption algorithms: DES, 3DES, and AES. The performance comparison hinges on the security level for the data, encryption time taken, and the performance required for the encryption process, particularly for varied input data.

[29] Blowfish has been found to be a better option than AES and Rijndael in an audio encryption study that compares all the above algorithms. Such a finding points towards a growing trend of securing audio streams in computer networks, where data transfer is given the utmost priority.

3. MATERIALS AND METHODS

The process includes the description of algorithms, the implementation of encryption and decryption methods, simulation, and performance evaluation of the chosen algorithms (Fig. 1).

3.1. Description of the Cryptographic Algorithms

3.1.1. Blowfish

Bruce Schneier created Blowfish, another symmetric key block cipher, in 1993. It became well-known because of its performance and adaptability. Blowfish works with 64-bit blocks and uses various key lengths ranging from 32 to 448 bits. Its architecture ensures speed and security with an effective Feistel network topology and a crucial expansion phase. Despite Blowfish's efficiency and security, its use has decreased in favor of more recent algorithms like AES [30].

Algorithm 1: Blowfish algorithm

```
STEP1: Start
STEP2: BLOCK_SIZE = 8
STEP3: NUM_ROUNDS = 16
STEP4: P [0.17]
STEP5: S [0. 3][256]
STEP6: for i from 0 to 17:
STEP7 P[i] = predefined_value[i]
STEP8: for i from 0 to 3:
STEP9: for j from 0 to 255:
STEP10: S[i][j] = predefined_s_box_values[i][j]
STEP11: key_length = length(key)
STEP12: j = 0
STEP13: for i from 0 to NUM_ROUNDS + 1:
STEP14: combined_key = combine_key_and_index(key, j)
STEP15: P[i] XOR= combined_key
STEP16 block = encrypt_block(0x00000000, 0x00000000)
STEP17: L, R: split(block)
STEP18: P [i * BLOCK_SIZE (i +1) * BLOCK_SIZE -1] = L || R
STEP19: For all S-boxes do the same as above.
STEP20: function encrypt_block (L, R):
STEP21: For round from 1 to NUM_ROUNDS do:
STEP22: L XOR= F(R)
STEP23: swap (L, R)
STEP24: swap (L, R)
STEP25: return (R || L)
STEP26: function F(R):
STEP27: A, B, C, D: byte extraction of R into four parts
STEP28: return ((S[0][A] + s [1][B]) XOR S [2][C]) + s [3][D]
STEP29: function combine_key_and_index(key,j):
STEP30: return key [j % length(key)]
STEP31: Stop.
```

3.1.2. Data Encryption Standard (DES)

One of the earliest popular symmetric key algorithms was the DES algorithm, which was created by IBM in the early 1970s and became a federal standard in 1977. DES uses a 56-bit key to operate on 64-bit data blocks. Despite its initial widespread use, increases in computing power have made DES susceptible to brute-force attacks. Because of this, its use has decreased in favor of safer substitutes. According to studies, DES is currently regarded as outdated for the majority of applications needing high security [31-40].

Algorithm 2: Data encryption standard algorithm

```
STEP1: Start
STEP2: function encrypt(sequence):
STEP3: for round in rounds [i = 1... 16]
STEP4: ciphertext = round.encrypt(ciphertext)
STEP5: endfor
STEP6: return final permutation on ciphertext
STEP7: endfunction
STEP8: function decrypt(sequence):
STEP9: for round in rounds [i = 16... 1]
STEP10: plaintext = round.decrypt(plaintext)
STEP11: endfor
STEP12: return initial permutation inverse on plaintext
STEP13: endfunction
STEP14: Stop
```

3.1.3. Advanced Encryption Standard (AES)

In 2001, the National Institute of Standards and Technology (NIST) standardized AES, which was created as DES's replacement. AES handles data in blocks of 128 bits and provides key sizes of 128, 192, and 256 bits, in contrast to DES. Because of its architecture, which is based on the Rijndael algorithm, AES is resistant to known cryptographic attacks since it prioritizes security, efficiency, and flexibility. From safeguarding private government information to securing financial transactions, AES is extensively utilized in many different applications [32].

Algorithm 3: Advanced encryption standard algorithm

```
STEP1: Start
STEP2: function AES_256_Encrypt (plaintext, key)
STEP3: roundKeys = KeyExpansion(key)
STEP4: state = AddRoundKey(plaintext, roundKeys[0])
STEP5: for i from 1 to 13:
STEP6: state = SubBytes(state)
STEP7: state = ShiftRows(state)
STEP8: state = MixColumns(state)
STEP9: state = AddRoundKey (state, roundKeys[i])
STEP10: state = SubBytes(state)
STEP11: state = ShiftRows(state)
STEP12: ciphertext = AddRoundKey (state, roundKeys [14])
STEP13: return ciphertext
STEP14: Stop.
```

3.2. Simulation and Settings

The three widely used encryption algorithms used in this study are AES, DES, and Blowfish. Select text file sizes of 0.5MB, 1MB, 2MB, 5MB, 10MB, and 20MB were chosen to test the simulation of the algorithms in a Python programming environment. The AES, DES, and Blowfish encryption algorithms were evaluated using metrics such as file size, throughput, execution time, and avalanche impact (Table **1**).

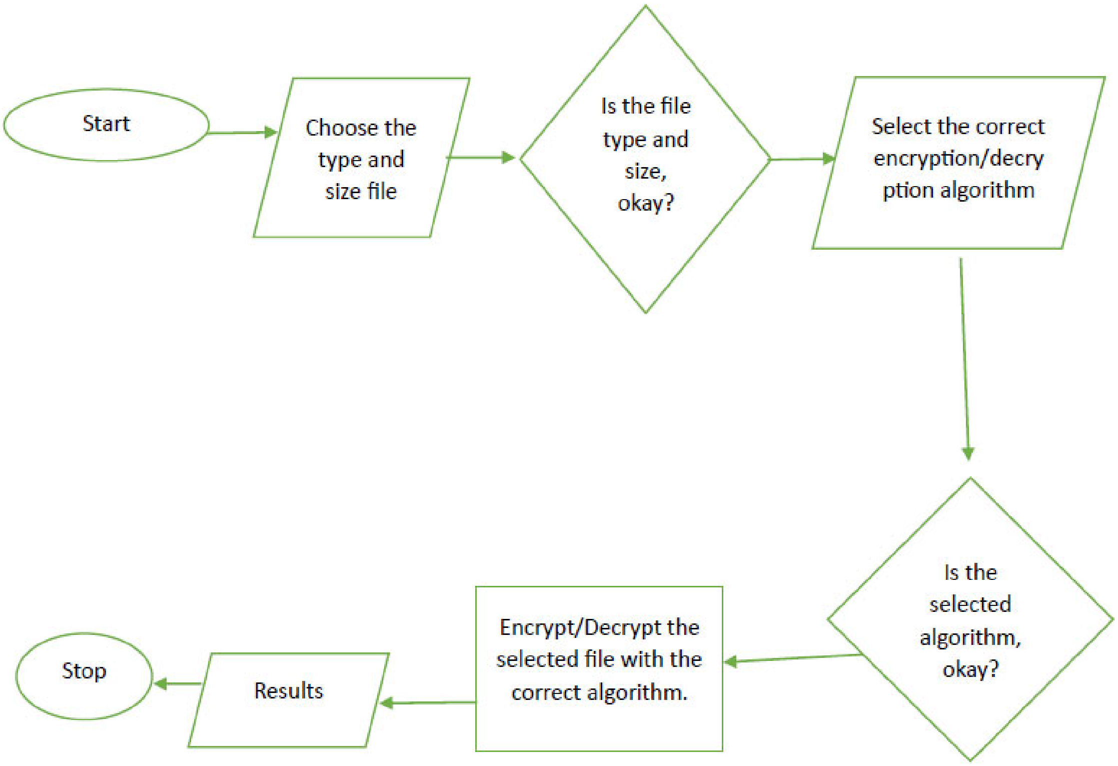
Start
Choose the type and size file
Is the file type and size, okay?
Select the correct encryption/decryption algorithm
Is the selected algorithm, okay?
Encrypt/Decrypt the selected file with the correct algorithm.
Results
Stop

Fig. (1). Process flowcharts for typical cryptographic algorithms.

Table 1. Algorithms' settings.

Algorithms	AES	DES	BLOWFISH
Block Size	128 bits	64 bits	64 bits
Key Size	256 bits	56 bits	32 bits
Number of Rounds	14 Rounds	16 Rounds	16 Rounds
Data Type	.txt	.txt	.txt
Data Size	0.5MB, 1MB, 2MB, 5MB, 10MB, and 20MB.	0.5MB, 1MB, 2MB, 5MB, 10MB, and 20MB.	0.5MB, 1MB, 2MB, 5MB, 10MB, and 20MB.

3.3. System Parameters

The system used for the study has an Intel® Core™ i7-7500U CPU running at 2.70 and 2.90 GHz, 16 GB of DDR4 RAM, and Windows 11 Pro 64-bit. The software environment included NumPy for numerical calculations and Python 3.8 with the PyCrypto module for performing encryption techniques.

3.4. Evaluation Metrics

The strengths and weaknesses of each encryption method are unique. Understanding an algorithm's performance, strengths, and weaknesses is necessary to apply an appropriate cryptography algorithm to a given application. Therefore, it is necessary to analyze encryption methods using several features. The following metrics were used in this study to assess the efficacy of the chosen encryption and decryption techniques:

3.4.1. Avalanche Effect Analysis

Avalanche measures the impact of small changes in the input data, *e.g.*, flipping a bit in the data, on the result of the encryption function and the consequent change in the ciphertext [33]. These three approaches have been implemented using the Python programming language with various plaintext sizes in an attempt to explore this very significant security parameter.

$$\text{Avalanche Test} = \frac{\text{Number of Changed bit in cipher text}}{\text{Number of bits in cipher text}} * 100\%$$

3.4.2. Execution Time Analysis

The time taken for encryption is vital for all but a few applications, and for embedded systems, this need is even more acute. For each of the techniques, the time for encryption is calculated by specifying an amount of data that is encrypted and, subsequently, decrypted [34].

$$Execution\ Time = Encryption\ Time + Decryption\ Time$$

3.4.3. Throughput Measurement

The throughput of each technique was determined by timing the encryption and decryption of each file. The following formula [35] was used to determine the throughput.

$$Throughput = \frac{File\ Size\ (MB)}{Encryption\ Time\ (secs)}$$

3.5. Sample Size and Statistical Justification

Three symmetric cipher algorithms, namely the AES algorithm, DES algorithm, and Blowfish algorithm, were analyzed in this research, using six different sizes for the plaintext files, viz., 0.5MB, 1MB, 2MB, 5MB, 10MB, and 20MB. These sizes are realistic enough to represent a reasonable class of file sizes used in communication-domain applications like messaging or document transfers. The objective was to evaluate the efficiency of the algorithms while changing the amount of data. To enable fair comparison of performance, multiple runs were made for each size with same parameters, achieving averages in terms of execution time, throughput, and impact too, and thereby reducing run-to-run variability in tests. That was achieved by repeating each test run.

Descriptive statistics in the form of means and standard deviations have been used to evaluate the trend of the performance of each algorithm with the various sizes of the files. However, inferential statistics such as the use of t-tests or ANOVA have not been applied here; the interest is only in the trend of the evolution of the performance. For more statistical robustness in the future, research with larger datasets is recommended. Instead of a random set of data, inferential statistics could be used to find out whether the variations in the algorithms' performances are statistically significant or not.

4. RESULTS AND DISCUSSIONS

This section shows the results of executing the simulation software for variations in text file sizes. It displays how the AES, DES, and Blowfish encryption algorithms vary with respect to changes in file size.

4.1. Avalanche Effect

Figure 2 demonstrates that AES generated the best avalanche effect across all file sizes, from 53.90% to 59.37%. The avalanche effect was consistent for DES at roughly 53.12% and more variable for Blowfish at 39.00% to 48.4%.

4.2. Execution Time

As illustrated in Fig. (3), the Blowfish algorithm had the quickest execution time across all text file sizes. Although AES did well, the execution times grew in direct proportion to the size of the file. DES took the longest to execute and was the slowest of them all.

4.3. Throughput

According to (Fig. 4), Blowfish has the highest throughput (1,634.5) for all text file sizes, followed by AES (250.9) and DES (151.3).

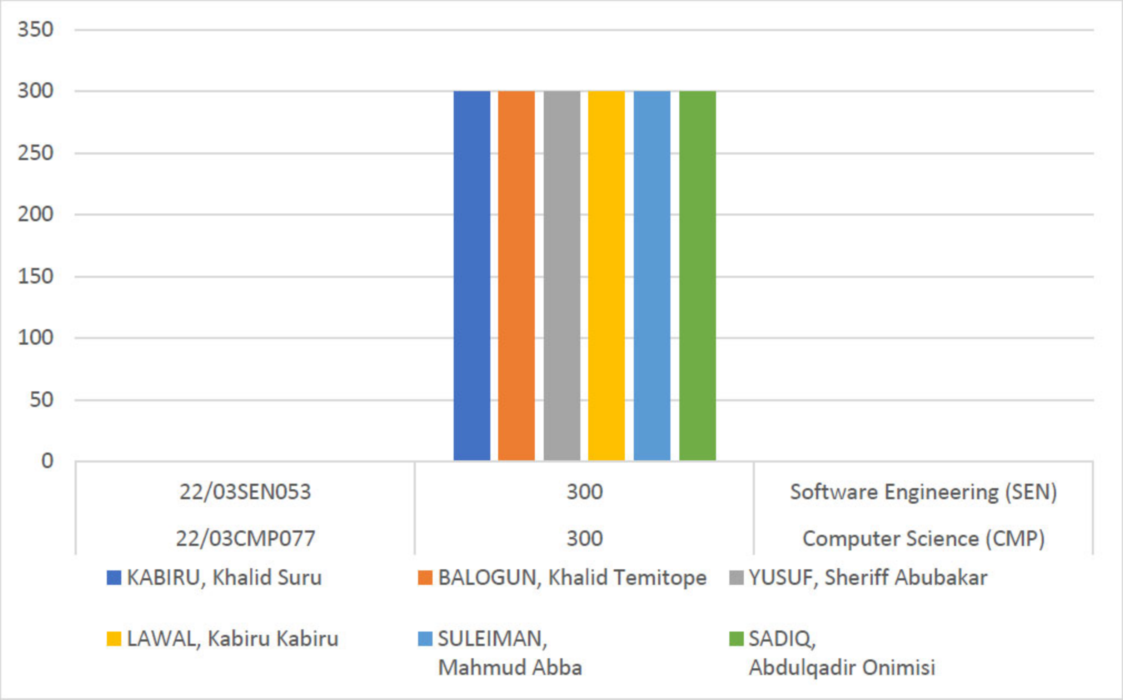

Fig. (2). Impact of avalanche on Blowfish, DES, and AES for varying text file sizes.

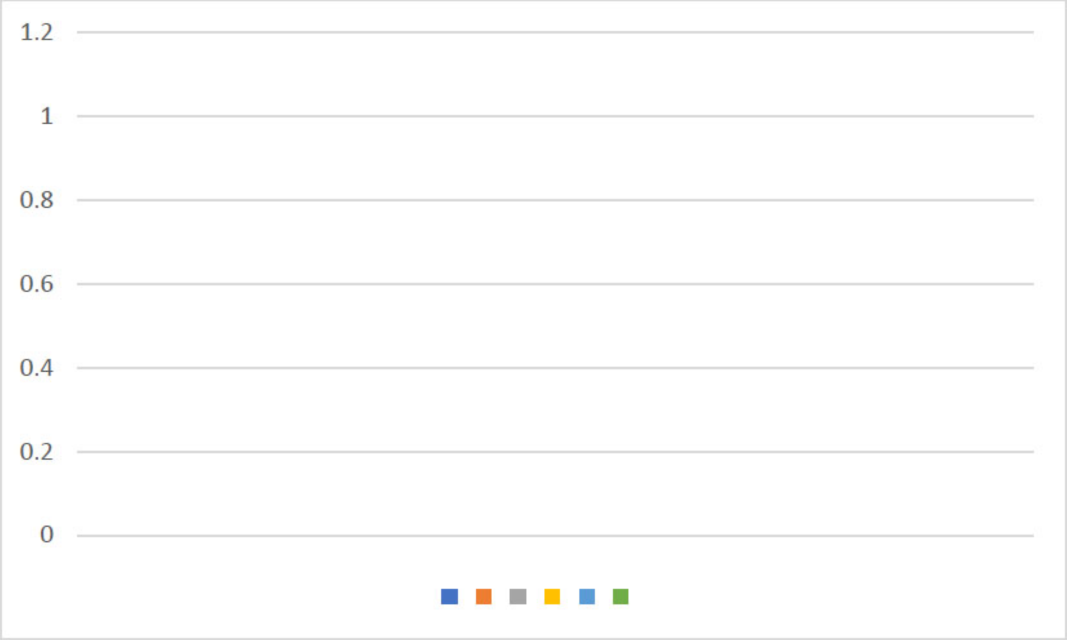

Fig. (3). AES, DES, and Blowfish execution times for various file sizes.

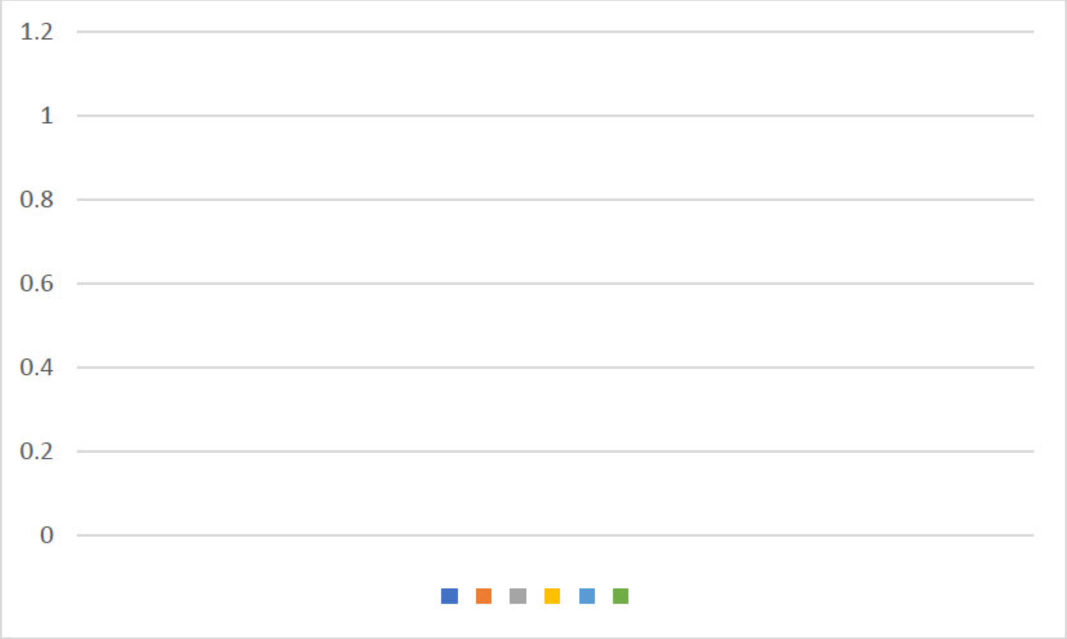

Fig. (4). DES, AES, and Blowfish throughput on file sizes.

CONCLUSION

The AES algorithm, DES algorithm, and Blowfish algorithm were rigorously tested through a Python implementation for different file sizes. From the results, it is clear that AES is more secure, but Blowfish performs extremely well due to its faster execution and higher throughput. In terms of security and efficiency, DES is not very good for current scenarios. However, it is extremely important and unique as it is one of the first comparisons of three significant symmetric algorithms for different file sizes, presenting a realistic scenario involving balanced performance, security, and applicability, which is surely a unique idea. It provides researchers and developers with an idea of which algorithms should be applied to different communication scenarios, based on parameters such as security and efficiency. Although it is unique, it is based on only text files and a few algorithms, which need to be further extended for validation for more multimedia files, large files, and even efficient algorithms for further validation and applicability.

RECOMMENDATIONS

The results show that symmetric encryption methods are applicable and comply with all set performance requirements. For high security needs, AES is chosen on account of the high effect of the avalanche and security attributes. For real-time applications, Blowfish is chosen because of its optimal speed. To conclude, AES must be considered for extremely security-critical situations, whereas Blowfish is best suited for situations where speed is of utmost significance.

FUTURE RESEARCH DIRECTIONS

This can be further investigated; even hybridization of Blowfish and AES can be considered in order to attain the best of both security and efficiency. Evaluation of its usage in various scenarios or applications, e.g., in IoT devices or resource-constrained applications, will be beneficial. The extended versions of those will therefore be even more beneficial for the creation of the upcoming ChaCha20 and Salsa20 algorithms, as well as quantum-resistant cryptography techniques, which are essential to strengthening this line of encryption against impending threats.

AUTHORS' CONTRIBUTIONS

The authors confirm their contribution to the paper as follows: A.A.: Study conception and design; O.R., O.A.: Methodology; A.A.: Writing - Reviewing and Editing; K.T.: Writing - Original Draft Preparation. All authors reviewed the results and approved the final version of the manuscript.

LIST OF ABBREVIATIONS

AES	=	Advanced Encryption Standard
DES	=	Data Encryption Standard
GUI	=	Graphical User Interface

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

The data and supportive information are available within the article.

FUNDING

None.

CONFLICT OF INTEREST

Dr. Oluwasegun Aroba is a member of the Editorial Advisory Board of the journal TOBEJ.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] R. Romansky, "Digital age and personal data protection", *Int. J. Information Technol. Security,* vol. 14, no. 3, pp. 89-100, 2022.

[2] A. Tewari, and B.B. Gupta, "Security of internet of things based on cryptographic algorithms: A survey", *Wirel. Netw.,* vol. 26, no. 2, pp. 1-20, 2022.

[3] K. Ermoshina, and F. Musiani, *Concealing for Freedom: The Making of Encryption, Secure Messaging and Digital Liberties.,* Mattering Press, 2022. [http://dx.doi.org/10.28938/9781912729227]

[4] B. Seth, S. Dalal, V. Jaglan, D.N. Le, S. Mohan, and G. Srivastava, "Integrating encryption techniques for secure data storage in the cloud", *Trans. Emerg. Telecommun. Technol.,* vol. 33, no. 4, p. e4108, 2022. [http://dx.doi.org/10.1002/ett.4108]

[5] Y. Xu, "Research on computer information network security technology and development direction", *J. Comput. Electronic. Inform. Manag.,* vol. 16, no. 2, pp. 21-24, 2025. [http://dx.doi.org/10.54097/zjqkkb50]

[6] S.K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari, "Security of internet of things based on cryptographic algorithms: A survey", *Wirel. Netw.,* vol. 27, no. 2, pp. 1515-1555, 2021. [http://dx.doi.org/10.1007/s11276-020-02535-5]

[7] M. Ghiasi, T. Niknam, and Z. Wang, "A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future", *Electr. Power Syst. Res.,* vol. 189, p. 106664, 2024. [http://dx.doi.org/10.1016/j.epsr.2022.108975]

[8] A. Razaque, "Performance evaluation and analysis of advanced symmetric key cryptography algorithms", *Int. J. Comput. Appl.,* vol. 182, no. 44, pp. 1-9, 2022.

[9] Q. Zhang, "An overview and analysis of hybrid encryption: the combination of symmetric encryption and asymmetric encryption", *2021 2nd International Conference on Computing and Data Science (CDS)* CA, USA, 2021, pp. 616-622 [http://dx.doi.org/10.1109/CDS52072.2021.00111]

[10] 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, p. 44, 2023. [http://dx.doi.org/10.14569/IJACSA.2023.0140344]

[11] J. Ahn, R. Hussain, K. Kang, and J. Son, "Exploring encryption algorithms and network protocols: A comprehensive survey of threats and vulnerabilities", *IEEE Commun. Surv. Tutor.,* p. 1, 2025. [http://dx.doi.org/10.1109/COMST.2025.3526605]

[12] S. Singh, P.K. Sharma, S.Y. Moon, and J.H. Park, "Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions", *J. Ambient Intell. Humaniz. Comput.,* vol. 15, no. 2, pp. 1625-1642, 2024. [http://dx.doi.org/10.1007/s12652-017-0494-4]

[13] B. Halak, Y. Yilmaz, and D. Shiu, "Comparative analysis of energy costs of asymmetric vs symmetric encryption-based security applications", *IEEE Access,* vol. 10, pp. 76707-76719, 2022. [http://dx.doi.org/10.1109/ACCESS.2022.3192970]

[14] D. Ramakrishna, and M.A. Shaik, "A comprehensive analysis of cryptographic algorithms: Evaluating security, efficiency, and future challenges", *IEEE Access,* vol. 13, p. 11576, 2024. [http://dx.doi.org/10.1109/ACCESS.2024.3518533]

[15] Y. Dodis, K. Haralambiev, A. LA3pez-Alt, and D. Wichs, "Cryptography against continuous memory attacks", Available from: https://eprint.iacr.org/2010/196

[16] R Iqbal, NM Ansari, M Ismail, and H Gul, "Design and evaluation of lightweight cryptographic algorithms for Internet of Things (IoT) devices: Achieving optimal trade-offs between security,

computational speed, and energy efficiency in resource-constrained environments", *J Multidiscip Stud,* vol. 6, no. 1, pp. 85-99, 2025. [http://dx.doi.org/10.71016/tp/smfybz24]

[17] A. Gour, S.S. Malhi, G. Singh, and G. Kaur, "Hybrid cryptographic approach: for secure data communication using block cipher techniques. InE3S Web of Conferences", *E3S Web Conf.,* vol. 556, p. 01048, 2024. [http://dx.doi.org/10.1051/e3sconf/202455601048]

[18] 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. Computer Sci. Mobile Comput.,* vol. 13, no. 4, pp. 68-87, 2024. [http://dx.doi.org/10.47760/ijcsmc.2024.v13i04.008]

[19] A.S. Yabo, M. Aliyu, S.G. Auyo, A. Ali, A.M. Haruna, and B.F. Rugga, "Comparative analysis of encryption algorithms using simulation technique", *Caliphate J. Sci. Technol.,* vol. 6, no. 1, pp. 109-112, 2024. [http://dx.doi.org/10.4314/cajost.v6i1.14]

[20] R. Umapathi, M. Rethinasabapathy, V. Kakani, H. Kim, Y. Park, H.K. Kim, G.M. Rani, H. Kim, and Y.S. Huh, "Hexagonal boron nitride composite film based triboelectric nanogenerator for energy harvesting and machine learning assisted handwriting recognition", *Nano Energy,* vol. 136, p. 110689, 2025. [http://dx.doi.org/10.1016/j.nanoen.2025.110689]

[21] R.K. Muhammed, R.R. Aziz, A.A. Hassan, A.M. Aladdin, S.J. Saydah, T.A. Rashid, and B.A. Hassan, "Comparative analysis of aes, blowfish, twofish, salsa20, and chacha20 for image encryption", *arXiv,* 2024.

[22] E.A. AL-Maqtari, "Performance evaluation for AES, blowfish, DES, and 3DES cryptography algorithms", *Partner. Univers. Innovat. Res. Citation.,* vol. 2, no. 5, pp. 86-95, 2024.

[23] A.T. Olumide, O.O. Oyelayo, O.O. Lawal, and S.G. Akinyemi, "Performance evaluation of some selected image encryption algorithms", *Ilorin J. Computer Sci. Inform. Technol.,* vol. 7, no. 1, pp. 65-76, 2024.

[24] SQ Yeow, and KW Ng, "Neural network based data encryption: A comparison study among DES, AES, and HE techniques", *Int. J. Inform. Visualization,* vol. 7, no. 3-2, pp. 2086-2094, 2023. [http://dx.doi.org/10.30630/joiv.7.3-2.2336]

[25] S. Sabeen, "Securing cloud data: A comprehensive review of hybrid cryptography and analysis of AES, Blowfish, and Twofish Algorithms", *3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)* 2024, pp. 568-575

[26] RP Saputra, J Wahyudi, and J Jumadi, "Comparative analysis of the Blowfish algorithm and the DES algorithm in the document file encryption and decryption process", *J Komputer Inform dan Teknologi,* vol. 2, no. 2, pp. 605-612, 2022. [http://dx.doi.org/10.53697/jkomitek.v2i2.1041]

[27] H. Dibas, and K.E. Sabri, "A comprehensive performance empirical study of the symmetric algorithms: AES, 3DES, Blowfish and Twofish", *2021 International Conference on Information Technology (ICIT)* Amman, Jordan, 2021, pp. 344-349 [http://dx.doi.org/10.1109/ICIT52682.2021.9491644]

[28] B.E.H.H. Hamouda, "Comparative study of different cryptographic algorithms", *Journal of Information Security,* vol. 11, no. 3, pp. 138-148, 2020. [http://dx.doi.org/10.4236/jis.2020.113009]

[29] C. Rathod, and A. Gonsai, "Performance analysis of AES, Blowfish and Rijndael: cryptographic algorithms for audio. InRising Threats in Expert Applications and Solutions", *Proceedings of FICR-TEAS,* vol. 2020, pp. 203-209, 2020. [Singapore: Springer Singapore.].

[30] A.E. Adeniyi, S. Misra, E. Daniel, and A. Bokolo Jr, "Computational complexity of modified blowfish cryptographic algorithm on video data", *Algorithms,* vol. 15, no. 10, p. 373, 2022. [http://dx.doi.org/10.3390/a15100373]

[31] N. Mishra, and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review", *IEEE Access,* vol. 9, pp. 59353-59377, 2021. [http://dx.doi.org/10.1109/ACCESS.2021.3073408]

[32] O.P. Olaiya, T.O. Adesoga, A.A. Adebayo, F.M. Sotomi, O.A. Adigun, and P.M. Ezeliora, "Encryption techniques for financial data security in fintech applications", *International Journal of Science and Research Archive,* vol. 12, no. 1, pp. 2942-2949, 2024. [http://dx.doi.org/10.30574/ijsra.2024.12.1.1210]

[33] H.T. Assafli, and I.A. Hashim, "Security enhancement of AES-CBC and its performance evaluation using the avalanche effect", *2020 3rd International Conference on Engineering Technology and its Applications (IICETA)* Najaf, Iraq, 2020, pp. 7-11 [http://dx.doi.org/10.1109/IICETA50496.2020.9318803]

[34] P. Pronika, and S.S. Tyagi, "Performance analysis of encryption and decryption algorithm", *Indones. J. Electr. Eng. Comput. Sci.,* vol. 23, no. 2, pp. 1030-1038, 2021. [http://dx.doi.org/10.11591/ijeecs.v23.i2.pp1030-1038]

[35] P. Panahi, C. BayŽñlmŽñ□Y, U. AØavu□YoŽYlu, and S. KaAar, "Performance evaluation of lightweight encryption algorithms for IoT-based applications", *Arab. J. Sci. Eng.,* vol. 46, no. 4, pp. 4015-4037, 2021. [http://dx.doi.org/10.1007/s13369-021-05358-4]

[36] O.J. Aroba, and M. Rudolph, "Systematic literature review on the application of precision agriculture using artificial intelligence by small-scale farmers in Africa and its societal impact", *J. Infrastruct. Polic. Develop.,* vol. 8, no. 13, p. 8872, 2024. [http://dx.doi.org/10.24294/jipd8872]

[37] O.J. Aroba, T. Xulu, N.N. Msani, T.T. Mohlakoana, E.E. Ndlovu, and S.M. Mthethwa, "The adoption of an intelligent waste collection system in a smart city", *2023 Conference on Information Communications Technology and Society (ICTAS)* Durban, South Africa, 2023, pp. 1-6 [http://dx.doi.org/10.1109/ICTAS56421.2023.10082750]

[38] O.J. Aroba, "The implementation of augmented reality in internet of things for virtual learning in higher education", *Int. J. Computing Sci. Res.,* vol. 8, pp. 2536-2549, 2024. [http://dx.doi.org/10.25147/ijcsr.2017.001.1.174]

[39] OJ Aroba, N Naicker, T Adeliyi, A Gupthar, and K Karodia, "A review: The bibliometric analysis of emerging node localization in wireless sensor network", *Int. J. Computer Inform. Syst. Industr. Manag. Appli.,* vol. 15, pp. 141-153, 2023.

[40] A.B. Sakpere, H.O. Aworinde, O.F. Afe, S. Adebayo, A.E. Adeniyi, and O.J. Aroba, "Exploring User Adoption and Experience of Automated Machine Learning Platforms with a Focus on Learning Curves, Usability, and Design Considerations", *Open Biomed. Eng. J.,* vol. 19, no. 1, p. e18741207395767, 2025. [http://dx.doi.org/10.2174/0118741207395767250818130213]

DISCLAIMER: The above article has been published, as is, ahead-of-print, to provide early visibility but is not the final version. Major publication processes like copyediting, proofing, typesetting and further review are still to be done and may lead to changes in the final published version, if it is eventually published. All legal disclaimers that apply to the final published article also apply to this ahead-of-print version.