WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM

e-ISSN: 2788-5879 p-ISSN: 2788-5879

Design and Implementation of Multi-key Blowfish and CAST Algorithm: Comparative Study with CBC, CFB and CTR Modes

Bushra Jaber M.Jawad[1,2,*] and Saif Mahmood Al-alak[3]

[1]Department of Information Networks, College of Information Technology, University of Babylon, Iraq
[2]Department of Accounting, College of Administration and Economics, University of Kerbala, Iraq
[3]Department of Computer Science, College of science for women, University of Babylon, Babel, Iraq

*Corresponding Author: Bushra Jaber M.Jawad

DOI: https://doi.org/10.52866/ijcsm.0000.00.00.000
Received: Augest 2023; Accepted: November 2023; Available online: December 2023

ABSTRACT: In modern communication systems, secret keys are used to secure sensitive information such as personal data, financial transactions and government secrets. However, the security of these systems is only as strong as the secrecy of the key utilized for encrypting and decrypting the data. The use of easy guessable or compromised keys can lead to unauthorized access, data breaches, and identity theft. Additionally, the distribution and management of secret keys present challenges such as key generation, key distribution, key revocation, and key renewal, which can be complex and costly. Therefore, finding secure and efficient methods to manage secret keys is crucial for ensuring the confidentiality, integrity, and availability of sensitive information in today's digital age. Therefore, to improve the security of the symmetric encryption algorithms, The new method to encrypt and decrypt the difference size files using multiple keys is proposed. In addition to investigate the impact of using multiple keys on the security and performance of symmetric algorithms. However, the multiple keys are generated previously using MD5 and SHA2. The chosen algorithms in this paper are Blowfish and CAST, with chosen three modes of operation: CBC, CFB and CTR. The presented results in the paper indicate that the Blowfish algorithm with using generated multiple keys outperforms in the CBC mode despite has the most encryption and decryption time consuming and memory usage. The randomness of Blowfish with CBC and CTR mode yield the highest throughput, then it is followed by Blowfish with CFB mode.

Keywords: Cryptography, symmetric algorithms, Secret keys, Blowfish algorithm, CAST algorithm

1. INTRODUCTION

Cryptography is an ancient practice used for centuries to safeguard communication from illegal access or tampering. It includes the use of mathematical algorithms to transform plain text into a cipher message, which can only be deciphered by those possessing the correct key. In the digital age, cryptography has become increasingly vital, especially for securing sensitive information transmitted online and stored in databases. It plays a critical role in various areas, such as online transactions, safeguarding personal data, and preserving national security secrets. Cryptography is an important aspect of current information security, applied in banking, e-commerce, and military communications, ensuring the protection and privacy of individuals and organizations amid the prevalence of cyber threats and data breaches [1, 2].

For a long time, symmetric algorithms have been preferred due to their efficiency and speed in securing data and communication. However, exclusively using a single secret key for the processes of encryption and decryption can introduce significant vulnerabilities. In this article, we will examine the weaknesses associated with using the same secret key in symmetric algorithms and discuss the potential risks involved [3].

Several widely recognized symmetric-key block ciphers that are presently accessible comprise DES, 3DES, CAST5, RC6, Blowfish, Twofish, Serpent, AES, TEA, IDEA, and MARS.

Blowfish algorithm is Developed by Bruce Schneier, Blowfish represents a symmetric 64 bit block cipher specifically designed for 32-bit processors equipped with generous data caches. The key length can vary between 32 and 448 bits, providing flexibility to users. Functioning as a 16 round Feistel cipher, Blowfish employs large key-dependent S-boxes that take 8 bit input and produce 32 bit output. In each round, one entry of the P-array is utilized, and upon completion of the final round, each half of the data block undergoes an XOR operation with one of the two remaining unused P-entries [4–6].

CAST, which represents Carlisle Adams and Stafford Tavares, is a symmetric key block cipher that was created in 1996. It has different varieties, specifically CAST-128, CAST-256, and CAST-512, each with particular block sizes and key lengths. CAST 128 works on 64-digit hinders and takes into consideration key lengths going from 40 to 128 pieces, though CAST-256 and CAST-512 are intended to help with bigger key lengths, offering upgraded safety efforts. The underlying structure of the CAST algorithm follows the Feistel network approach, utilizing S-boxes and key mixing operations to accomplish its encryption and decryption functions. CAST has gained acknowledgment for its robust security characteristics and has found utility in diverse applications that demand high-performance symmetric encryption [7, 8].

Symmetric key encryption involves the use of a single key for both encrypting and decrypting data blocks. However, this approach can become vulnerable to attacks when the same key is employed for many blocks, as attackers can exploit data patterns to hack the message. To overcome this security concern and prevent identical cipher texts for identical plaintexts, an extra input is incorporated into each encrypted block. This concept is referred to as block cipher modes of operation, which aim to bolster the security of symmetric key algorithms. Several block cipher modes exist, such as Electronic Code Book (ECB), Cipher Block Chaining (CBC), Plaintext or Propagating Cipher Block Chaining (PCBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). Each mode has its strengths and weaknesses. The Cipher Block Chaining mode (CBC) presents an element of randomness to thwart predictability-based attacks. The plain text is partitioned into fixed-size blocks, and each block, except the first one, is combined with the cipher text block of the preceding block using XOR. The first block is combined with a random initialization vector (IV) of the same length as the plaintext. This IV is a unique number used only once (nonce) and can be generated through a random number generator or counter. If a single-bit error occurs in the plaintext, recovering the original plain text from the cipher text becomes extremely difficult. However, a single-bit error in the cipher text would only impact the two consecutive plaintext blocks. It is crucial to note that encryption cannot be parallelized because each block's cipher text directly influences the encryption of the following block [9].

The Cipher Feedback (CFB) mode operates similarly to a stream cipher, In this process, the current block's plaintext is combined with the cipher text of the preceding block. The encryption process is the same for both encryption and decryption. The first block's plaintext is combined with an initialization vector using XOR to produce the cipher text. Decryption can be parallelized, but encryption cannot be done in parallel. Like other modes, if a single-bit error occurs in the plaintext, it will corrupt the entire cipher text. However, if a single-bit error occurs in the cipher text, only the following two blocks of plaintext would be affected. Similar to a stream cipher, the Counter (CTR) mode encrypts data using an additional input formed by combining an ascending counter and a nonce value. The length of the nonce initialization vector is shorter than the block length, and the counter is typically initialized starting from 0. This mode has gained widespread adoption due to its effectiveness. In CTR mode, encryption and decryption operations can be executed in parallel, which further contributes to its appeal [10].

The objectives of the paper are: to improve the security of symmetric encryption algorithms, we propose to use the generated multiple keys for encryption and decryption processes and investigate the impact of using multiple keys on the security and performance of symmetric algorithms.

Our contributions are: to analyze the trade-off between security and efficiency when using multiple keys in symmetric algorithms. In addition, to compare the performance and security of symmetric algorithms with multiple keys.

The rest of the paper is organized as follows: Section 2 discusses related work; while Section 3 details the proposed algorithm. Section 4 provides a detailed description of the results of experiments and analysis. Finally, in Section 5 presents the conclusion of proposed algorithm.

2. LITERATURE REVIEW

In [11], Awotunde el at. conducted an investigation involving four algorithms (DES, 3DES, AES, Blowfish) to assess various factors such as key size, memory usage, CPU usage time, and encryption speed. The aim was to determine the computational resources required and the time taken by each algorithm to complete its encryption task. Their findings revealed a direct relationship between the key length used in encryption algorithms and resource consumption in most cases. Notably, the (Blowfish) algorithm stood out as it consumed more time, memory, and CPU resources during encryption operations due to its utilization of a key length much higher than 448 bits. Furthermore, the study cautioned

against recommending high key length encryption algorithms for memory and power-sensitive devices, especially those small in size, as they may not perform efficiently under such conditions. These devices may experience challenges in handling the increased computational demands imposed by these algorithms.

In [12] the authors focus on identifying five commonly used block ciphers in Cipher Block Chaining (CBC) mode. Initially, they perform a multi-class identification of these five ciphers. From the results, it becomes evident that high accuracy in identifying the ciphers can only be achieved when both the keys and initialization vectors (IVs) are identical for both the training and testing cipher text files. To address the issue of a low identification rate when using different IVs or keys for training and testing cipher text, the authors proceed with a one-to-one identification approach between AES and the other four block ciphers.

The authors in this study showcased the outcomes of implementing and analyzing various cryptographic algorithms, including DES, 3DES, AES, RSA, and Blowfish. Additionally, they conducted comparisons among these cryptographic techniques, evaluating their performance, strengths, and weaknesses. [13]

In [14], an encryption/decryption technique known as Simple Lightweight and Highly Secure Encryption Decryption (SHSED) was presented. This method offers a high level of security and can be used in various data processing applications, including Cloud-based applications. The SHSED method employs straightforward and efficient logical operations like XORing, addition, subtraction, and byte shifting. The results obtained from applying this introduced method were then compared to the results of other encryption methods such as LED, AES, and DES.

Whereas, in [15], the authors conducted a comprehensive comparison of common symmetric algorithms, which are Blowfish, Twofish, AES and DES. The primary focus of the study was to assess the performance of these algorithms under different configurations and data sizes. The evaluation criteria included block size, key size and speed. Through simulations, the study found that Twofish outperformed the other algorithms in terms of performance. It demonstrated superior efficiency and effectiveness compared to the rest. Additionally, Twofish was considered a standard encryption algorithm due to its robust security, as it showed no known weaknesses at the time of evaluation. On the other hand, AES exhibited relatively weak performance results when compared to the other algorithms. Its encryption and decryption processes require more processing capacity, making it less efficient in certain scenarios. In summary, the comparison highlighted Twofish as the top-performing algorithm with strong security attributes, while AES showed weaker performance due to its higher processing demands.

Three encryption algorithms, AES (Advanced Encryption Standard), Blowfish, and Twofish are compared in this study. Metrics like throughput and encryption and decryption times are utilized in the comparison. According to the study, Twofish is a more effective data encryption technique for wireless networks than AES and Blowfish [16].

In [17], the authors proposed a system with cryptographic effects that comprise both robust security and simple performance analysis. To achieve more efficient data processing, the proposed system incorporates GPU assistance for handling huge amounts of data. A thorough comparison is made between the proposed algorithm and other symmetric cryptographic algorithms such as DES, AES, 3-DES, MARS and RC6. The evaluation is based on various criteria, including flexibility, architecture, security level and scalability, as well as throughput for both encryption and decryption processes and execution time. Additionally, the avalanche effect of the proposed system is also calculated and analyzed.

The study provided a comprehensive survey of common cryptographic algorithms. Ten different symmetric encryption algorithms, namely Blowfish AES, RC6, RC4, RC2, DES, XTEA, DESede, SEED and IDEA, were selected for a simulation. Various file sizes ranging between 1MB and 1GB were used to evaluate CPU utilization rate, encryption throughput and speed. The results were thoroughly analyzed, taking into account the level of security provided by each algorithm. The results were meticulously analyzed, considering the level of security offered by each algorithm. The research presents valuable insights into the performance and security aspects of these symmetric encryption algorithms, facilitating a better understanding of their suitability for different encryption needs and applications. [18]

This paper conducts a comparative analysis of performance among three prominent encryption algorithms: DES, 3DES and AES. The evaluation is centered around their data security capabilities, encryption time, and required throughput. The study highlights how the performance of these algorithms varies based on different input scenarios. [19]

In [20], the researchers performed a comparison of the sex algorithms: Blowfish, Twofish, DES, 3DES, AES and RC4 using the mode operation cipher block chaining (CBC), depending on two evaluation criteria: encryption and decryption time. Their results indicated that the 3DES algorithm depleted the most time through the processes of encryption and decryption, followed by DES. Conversely, RC4 exhibited the least execution time, with AES falling next in line. Twofish and Blowfish were positioned between these two levels in terms of processing time. Regarding the criterion of randomness, 3DES performed the best compared to the other algorithms. However, RC4 and AES scored poorly in this aspect, making them the least favorable choices concerning randomness.

Dibas et al. carried out a performance evaluation of four symmetric encryption algorithms: AES, 3DES, Blowfish and Twofish. Their assessment focused on various aspects, including execution time, text size and memory usage during encryption and decryption operations. In the results, they observed that Twofish exhibited the longest execution time for

both encryption and decryption processes, whereas AES displayed the shortest execution time. 3DES and AES algorithms consumed less memory than Twofish and Blowfish, with both AES and 3DES using nearly the same amount of memory. Moreover, AES employed less memory for decryption as well. In conclusion, Blowfish and Twofish were found to yield the largest cipher text sizes in comparison to the other algorithms. [21]

In this study, authors compare the encryption speeds of five cryptographic symmetric block-cipher algorithms: DES, 3DES, Blowfish, Twofish, and Threefish. To conduct the comparison, the findings demonstrate that Blowfish outperforms all the other tested algorithms in terms of encryption speed. [22]

The authors conducted a comparative assessment of two cryptography algorithms, namely the Advanced Encryption Standard (AES) and the Rivest Shamir Algorithm (RSA). Their goal was to determine which algorithm was more dependable based on factors such as cipher length, key length and encryption and decryption time. The results indicate that AES outperforms RSA, making it the superior encryption technique in this evaluation. [23]

Most of the researchers above only provide comparisons between encryption algorithms and do not provide improvements to these algorithms. Therefore, in this paper, we previously proposed an algorithm for deriving multiple keys that increases the robustness and performance of symmetric encryption algorithms.

3. PROPOSED ALGORITHM

To achieve confidentiality and improve the security of data and networks. Firstly, from our previous proposed algorithm, in which multiple keys are derived using MD5 and SHA2 algorithms and keys are stored in binary files. Then use these keys in encryption and decryption processes of different size files. The selected symmetric algorithms are Blowfish and CAST, as well as, the three operation modes: CBC, CFB and CTR.

After choosing the algorithm and operation mode, read the file that contains plain text and read the keys from the file. Furthermore, the parameters required in the selected mode will be entered, then the file will be encrypted and decrypted with the previously suggested and generated multiple keys.

After that, the performance of these algorithms will be calculated by the following metrics: encryption/decryption time, memory usage and throughput of encryption and decryption, and then evaluating these algorithms according to the resulting performance.

As shown in figure 1 the message or plain text is divided into blocks and each block will be encrypted using one key from the suggested multiple keys. the file contains plain text after reading and converting it to binary data. Then will be divided into many blocks. The size of the block differed from one to another algorithm and proposed algorithm. Thereafter, each block encrypts using one secret key from the multiple generated keys.

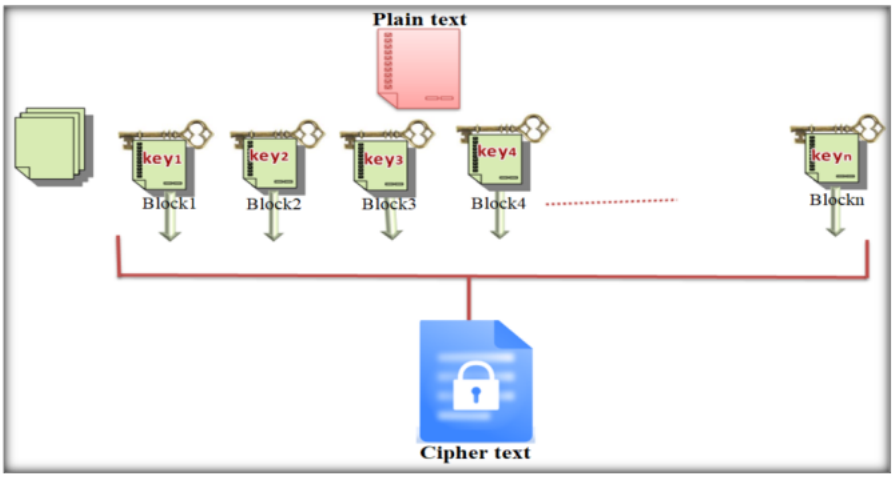

FIGURE 1. Diagram of the suggested algorithm

The equations (1) and (2) are the proposed encryption algorithm

$$Ci = E(ki, Pi) \tag{1}$$

$$Pi = D(ki,\ Ci) \tag{2}$$

Where P is the plain text, C is the cipher text and K is the key.

Encryption time refers to the duration taken by an encryption algorithm to convert plaintext into cipher ext. This time measurement excludes file I/O time. On the other hand, decryption time is the duration required by a decryption algorithm to transform cipher text back into plaintext. Similarly, this time measurement does not include file I/O time. Meanwhile, the throughput of an encryption scheme indicates the encryption speed and is calculated as the total amount of plaintext encrypted in kilobytes divided by the encryption time (KB/sec). As the throughput increases, power consumption decreases. Furthermore, Memory Utilization denotes the amount of memory utilized during the encryption or decryption process. [24].

To measure execution time or as it was called in our paper, encryption time, the method of time module time.time() is used to get the current time in seconds after importing the module time that allows it to work with time.

The time.time method is called to get the current time and it is assigned to a variable, it will be named “start time ” when the encryption process steps start. Then we call the same method again at the end of execution (the end of encryption possess) and assign it in another variable that indicates the end of the encryption process, then we subtract the end time from the start time to get the execution time or the encryption time, and the same steps are followed to get the decryption time. Thus to get a more accurate measurement, repeat the encryption process multiple times and calculate the average time. This helps account for variations in system load and other factors. The encryption time and decryption time can be expressed as in the equations (3) and (4) respectively.

$$Encryption\ Time = \frac{Total\ time\ for\ encryption}{Number\ of\ encryption\ operation} \tag{3}$$

$$Decryption\ Time = \frac{Total\ time\ for\ decryption}{Number\ of\ decryption\ operation} \tag{4}$$

The equations (5) and (6) are to measure encryption and decryption throughput respectively.

$$Encryption\ Throughput = \frac{Total\ encrypted\ text\ (byte)}{Encryption\ time} \tag{5}$$

$$Decryption\ Throughput = \frac{Total\ decryption\ text\ (byte)}{Decryption\ time} \tag{6}$$

Every encryption technique requires a particular and distinct memory size for its implementation. The memory needed is influenced by many factors such as the type of operations, the file size, the size of the adopted key and initialization vectors. For an algorithm to be considered effective, it must have a very small memory requirement [25].

Memory Utilization or Memory Usage defines how much memory is being consumed while doing the encryption or decryption processes. And to retrieve the memory usage of the current process. The object “memory_info()” is used with the attribute “.rss” for process “os.getpid()” by importing the library “psutil

Moreover, the figure 3 illustrated the three modes of operation and using the proposed multiple keys. Where IV represents an initialization vector, Pi is the plain text, Ci is the cipher text, and Ki are proposed multiple keys, which represent K1, K2, ... , Kn.

4. RESULTS AND DISCUSSION

4.1 PROGRAMMING ENVIRONMENT

The implementation of the symmetric algorithms is done by a Laptop with a core processor Intel i7, RAM 16 GB, and the operating system Windows 11Pro. The application is developed on Python with an integrated development environment (IDE) PyCharm.

4.2 RESULTS OF THE SIMULATION

This section is a study and analysis of the results of the proposed algorithm, the algorithm is suggested to achieve confidentiality for sensitive information and increase security through encrypting and transfer data between parity.

4.3 METHODOLOGIES

To analyse and evaluate the symmetric algorithms, three files have different sizes and three files of secret keys with different sizes are used to test the symmetric algorithms. In this paper, the implementation of two symmetric algorithms is suggested to encrypt the plain text in three files with different sizes and three files of keys.

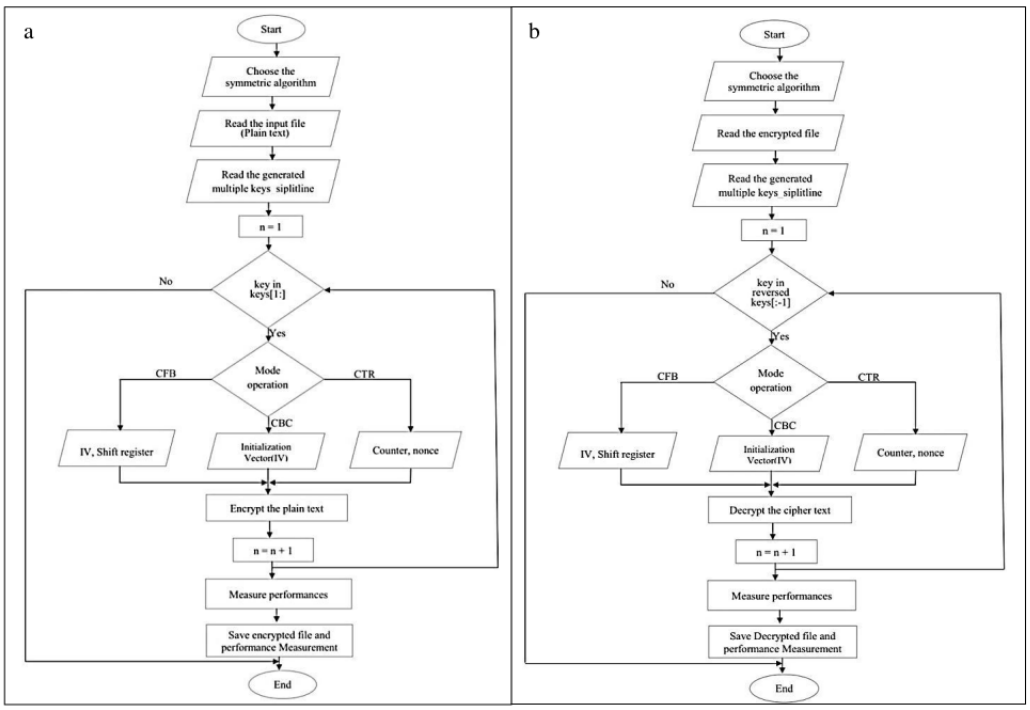

FIGURE 2. (a) Encryption and process; (b) - Decryption process

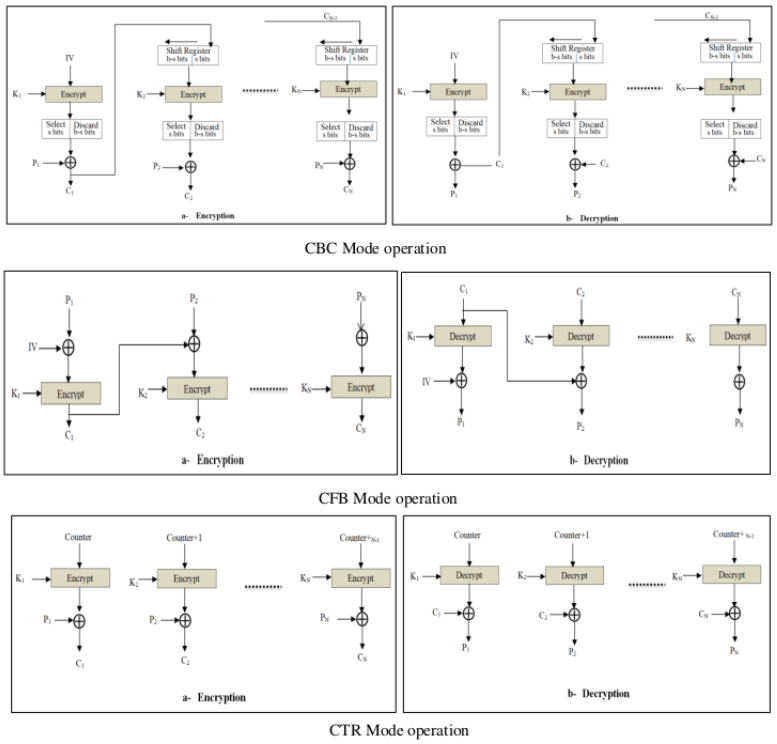

FIGURE 3. Three modes of operation CBC, CFB and CTR with proposed multiple keys

Table 1. The values of the parameters and metrics

Symmetric algorithms:	Blowfish, CAST
Number of files (plain text):	3 files with different sizes, 100 kb, 200kb and 1Mb
Number of keys:	3 files with different key sizes, 16, 32 and 128 bits with file size 10Mb, 15Mb, 30Mb
Metrics:	Encryption time, Decryption Time, Memory usage of an encryption and decryption, Throughput of an encryption and decryption and Randomness

4.4 EXPERIMENTS

There are two symmetric algorithms selected to implement in this study. Using multiple secret keys that generate according to the previous proposed algorithm. This proposed algorithm uses SHA2 and MD5 algorithms to generate multiple keys. However, the result is saved as a binary file or text file, including the random keys.

The experiments comprise Blowfish algorithm and CAST algorithm using three Mode operations: CBC, CFB, and CTR. The experiments also include different files (plain text) and three files that contain generated keys.

4.4.1. TIME ENCRYPTION AND DECRYPTION

As shown in Figure 4. The third file of plain text and the third file of the largest key in mode CBC will take the longest encryption and decryption times. But as it is clear, in general, when using all keys, the CAST and Blowfish algorithms consume encryption time with the third file.

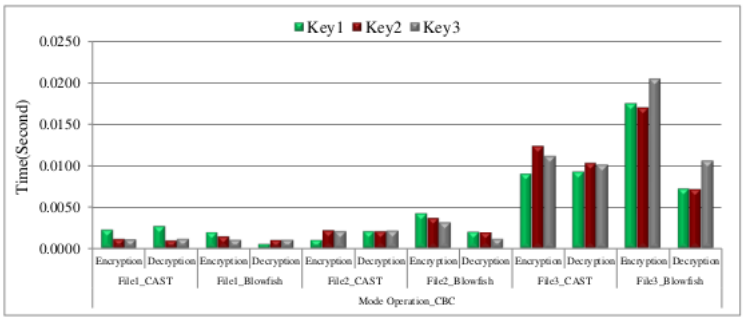

FIGURE 4. Encryption/decryption time of Blowfish and CAST algorithms with mode CBC

Likewise, in the figure 5 the Blowfish algorithm with mode CFB has the highest encryption and decryption time.

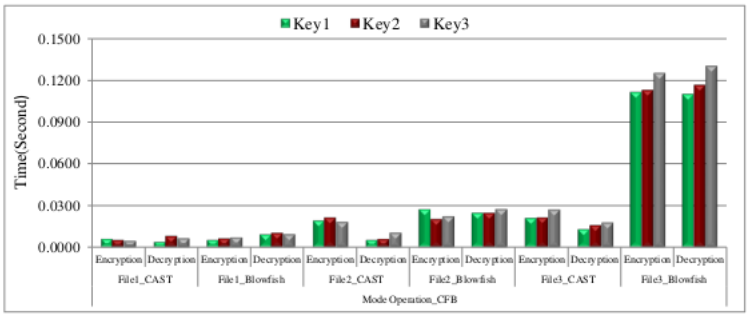

FIGURE 5. Encryption/decryption time of Blowfish and CAST algorithms with mode CFB

In Figure 6, the CAST algorithm with mode CTR consumes more encryption and decryption time.

4.4.2. MEMORY USAGE:

As shown in the figure 7. The mode CBC, it can be seen that there are not many differences except for a slight increase in the third key with the CAST algorithm with three files.

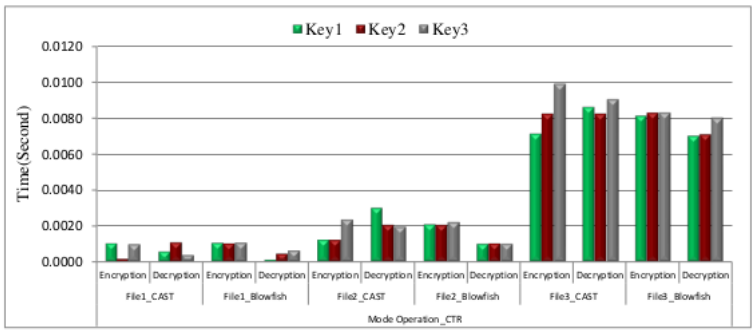

FIGURE 6. Encryption/decryption time of Blowfish and CAST algorithms with mode CTR

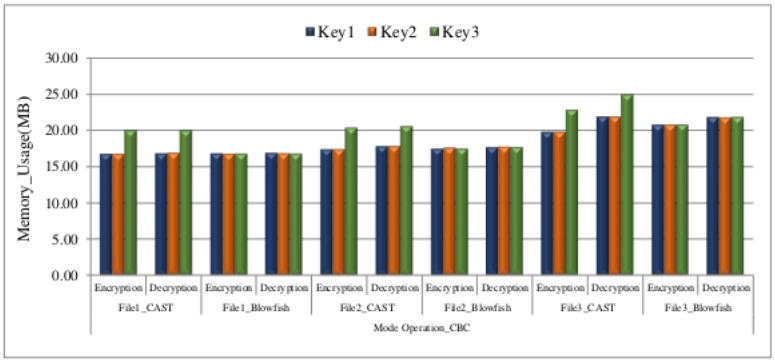

FIGURE 7. Encryption/decryption memory usage of Blowfish and CAST algorithms with mode CBC

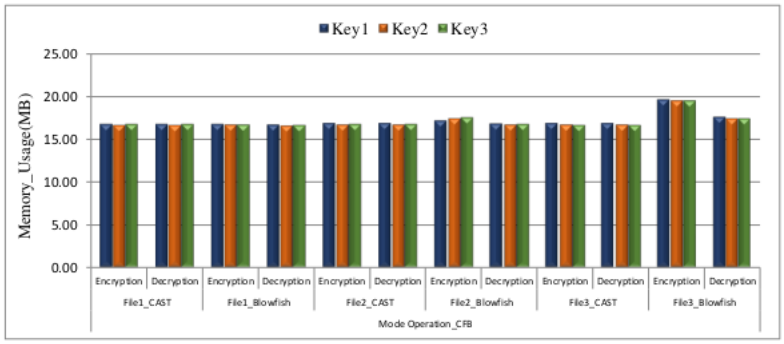

FIGURE 8. Encryption/decryption memory usage of Blowfish and CAST algorithms with mode CFB

As it is clear in figure 8. The Blowfish algorithm with mode operation CFB occupies the highest rank in memory, especially in encryption. Whereas in figure 9. With mode CTR, there is a great similarity between the two algorithms in memory usage for the same files and keys.

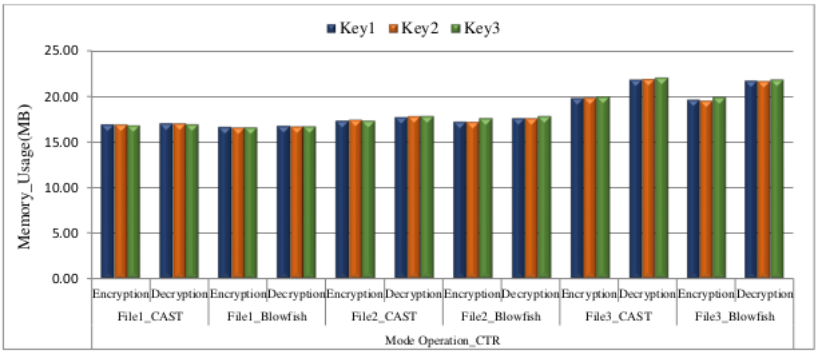

FIGURE 9. Encryption/decryption memory usage of Blowfish and CAST algorithms with mode CTR

4.4.3. THROUGHPUT ENCRYPTION AND DECRYPTION

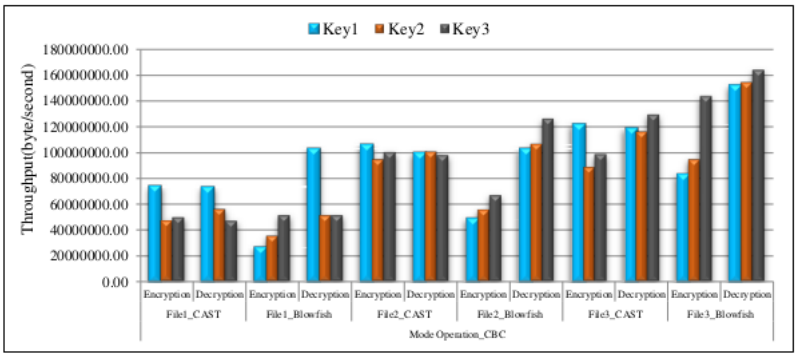

FIGURE 10. Encryption/decryption throughput of Blowfish and CAST algorithm with mode CBC

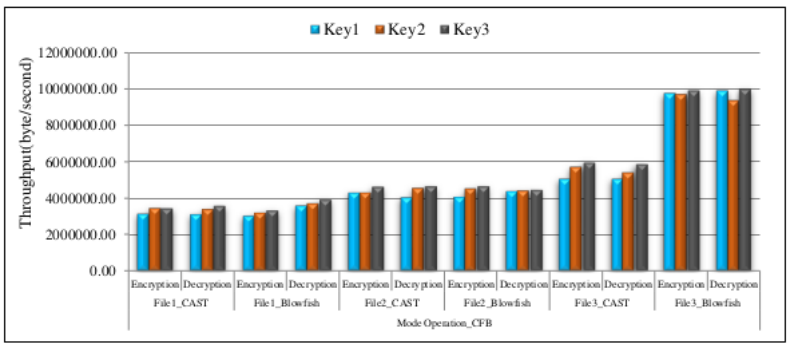

FIGURE 11. Encryption/decryption throughput of Blowfish and CAST algorithm with mode CFB

As shown in three figures 10, 11, 12, the Blowfish has the best throughput in three mode operations CBC, CFB and CTR, with the largest size file and the proposed multiple keys mentioned previously.

4.4.4. RANDOMNESS

Figures 13 and 14 present the randomness measurement results for the symmetric algorithms chosen in this study, namely Blowfish and CAST. Using the NIST Randomness Test suit [26]. As is well known, the strength of encryption

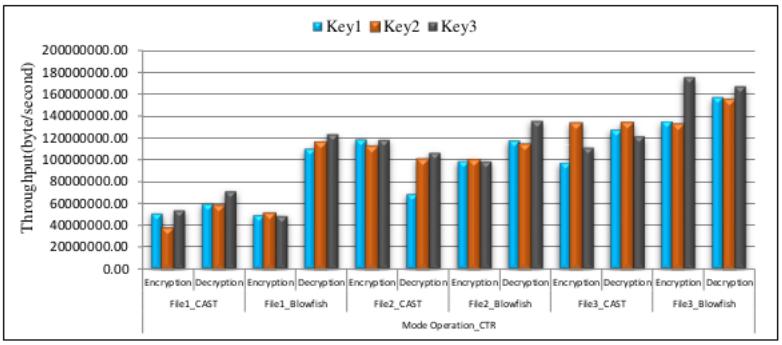

FIGURE 12. -Encryption/decryption throughput of Blowfish and CAST algorithm with mode CTR

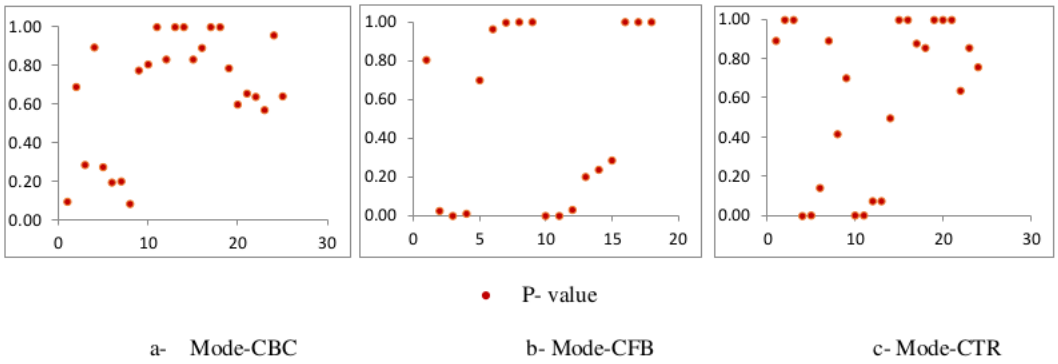

FIGURE 13. Randomness of the CAST algorithm

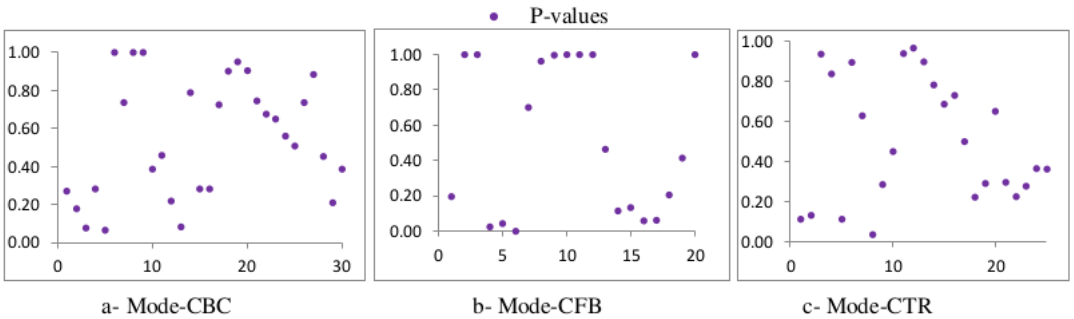

FIGURE 14. Randomness of the Blowfish algorithm

algorithms lies in measuring the randomness of the encrypted text by these algorithms. When we compare the results according to the randomness criterion in [20], the results show that Blowfish contained the most P-values within the safety region. In addition, the least number of p-values within the doubt region with modes CBC and CTR compared with the CAST algorithm, while some numbers of p-values within the failure region for both proposed algorithms with mode operation CFB.

Thus, we conclude from the above results that the chosen algorithms, which are the Blowfish and CAST algorithms, when used with traditional secret keys, such as key 1, don't take much time for encryption and decryption processes and don't consume memory. On the other hand, when the proposed algorithm is used to generate multiple keys, it will consume a lot of time for encryption and decryption operations. It also consumes memory, but in return, there will be high throughput compared to a traditional secret key. Additionally, when the randomness of the algorithms increases, their robustness increases and their resistance to attacks, especially attacks related to weak secret keys, such as brute force attacks, dictionary attacks and man-in the-meddle attacks.

5. CONCLUSION

The purpose of this paper is to assess how the symmetric algorithms can strengthen and improve the security by using the generated multiple keys, Algorithms SHA2 and MD5 are used with random numbers to generate the multiple secret keys. After that, compare three modes CBC, CFB and CTR for two selected algorithms which are Blowfish and CAST. Then measure the performance of these algorithms and randomness. The results of comparative study showed two chosen symmetric algorithms, Blowfish and CAST with CBC and CTR have high randomness. While the mode CFB has low randomness. However, the Blowfish algorithm is the best in both performance and randomness.

FUNDING

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] F. Maqsood, M. Ahmed, M. Mumtaz, and M. Ali, "Cryptography: A Comparative Analysis for Modern Techniques," *Int. J. Adv. Comput. Sci. Appl*, vol. 8, no. 6, 2017.
[2] B. Fish, E. Curve, B. Fish, and E. Curve, "A Comparative Study of Cryptographic Algorithms," *Int. J. Innov. Res. Multidiscip. F*, vol. 1, no. 2, pp. 24–28, 2015.
[3] Z. C. Oleiwi, W. A. Alawsi, W. C. Alisawi, A. S. Alfoudi, and L. H. Alfarhani, "Overview and Performance Analysis of Encryption Algorithms," *J. Phys. Conf. Ser*, vol. 1664, no. 1, pp. 0–15, 2020.
[4] E. Elgeldawi, M. Mahrous, and A. Sayed, "A Comparative Analysis of Symmetric Algorithms in Cloud Computing: A Survey," *Int. J. Comput. Appl*, vol. 182, no. 48, pp. 7–16, 2019.
[5] V. Parihar and M. A. Kulshrestha, "Blowfish Algorithm: a Detailed Study," *Int. J. Technol. Res. Eng*, vol. 3, no. 9, 2016.
[6] P. Patil, P. Narayankar, D. G. Narayan, S. M. Meena, . Des, A. 3des, and Blowfish, "A Comprehensive Evaluation of Cryptographic Algorithms," *Procedia Comput. Sci*, vol. 78, pp. 617–624, 2015.
[7] M. Ebrahim, S. Khan, and U. B. Khalid *Symmetric Algorithm Survey: A Comparative Analysis*, vol. 61, pp. 12–19, 2014.
[8] G. N. Krishnamurthy, V. Ramaswamy, G. H. Leela, and M. E. Ashalatha, "Performance enhancement of Blowfish and CAST-128 algorithms and Security analysis of improved Blowfish algorithm using Avalanche effect," *Int. J. Comput. Sci. Netw. Secur. IJCSNS*, vol. 8, no. 3, pp. 244–250, 2008.
[9] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication," *NIST Spec. Publ. 800-38B*, pp. 25–25, 2005.
[10] D. Bujari and E. Aribas, "Comparative analysis of block cipher modes of operation," *Int. Adv. Res. Eng. Congr*, pp. 2–5, 2017.
[11] J. B. Awotunde, A. O. Ameen, I. D. Oladipo, A. R. Tomori, and M. Abdulraheem, "Evaluation of four encryption algorithms for viability, reliability and performance estimation," *Niger. J. Technol. Dev*, vol. 13, no. 2, pp. 74–74, 2017.
[12] C. Tan, X. Deng, and L. Zhang, "Identification of block ciphers under CBC mode," *Procedia Comput. Sci*, vol. 131, pp. 65–71, 2018.
[13] N. A. Wahid, A. Ali, B. Esparham, and M. Marwan, "A Comparison of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish for Guessing Attacks Prevention," *J. Comput. Sci. Appl. Inf. Technol*, vol. 3, no. 2, pp. 1–7, 2018.
[14] A. Y. Hendi, M. O. Dwairi, Z. A. Al-Qadi, and M. S. Soliman, "A novel simple and highly secure method for data encryption-decryption," *Int. J. Commun. Networks Inf. Secur*, vol. 11, no. 1, pp. 232–238, 2019.
[15] R. Venkateshwarlu *Comparison of DES , AES , Blowfish and Twofish Symmetric Key Cryptography Algorithms*, vol. 6, pp. 639–647, 2019.

[16] A. Ghosh, "Comparison of Encryption Algorithms : AES , Blowfish and Twofish for Security of Wireless Networks," *Int. Res. J. Eng. Technol*, pp. 4656–4659, 2020.
[17] J. Prabhu and A. S. Dass, "Hybrid coherent encryption scheme for multimedia big data management using cryptographic encryption methods," *Int. J. Grid Util. Comput*, vol. 11, no. 4, pp. 496–496, 2020.
[18] M. N. Alenezi, H. Alabdulrazzaq, and N. Q. Mohammad, "Symmetric encryption algorithms: Review and evaluation study," *Int. J. Commun. Networks Inf. Secur*, vol. 12, no. 2, pp. 256–272, 2020.
[19] H. B. E. H. Hamouda, "Comparative study of different cryptographic algorithms," *J. Inf. Secur*, vol. 11, no. 4, pp. 138–148, 2020.
[20] M. T. Hadi and S. Al-Alak *Symmetric Security Algorithms Implementation in CBC Mode*, vol. 29, pp. 126–144, 2021.
[21] H. Dibas and K. E. Sabri, "A comprehensive performance empirical study of the symmetric algorithms:AES, 3DES, Blowfish and Twofish," *Conf. Inf. Technol. ICIT 2021 - Proc*, pp. 344–349, 2021.
[22] H. Alabdulrazzaq, M. N. Alenezi, . Des, 3des, T. Blowfish, and Threefish, "Performance Evaluation of Cryptographic Algorithms," *Int. J. Commun. Networks Inf. Secur*, vol. 14, no. 1, pp. 2022–2022.
[23] A. Olutola and M. Olumuyiwa, "Comparative Analysis of Encryption Algorithms," *Eur. J. Technol*, vol. 7, no. 1, pp. 1–9, 2023.
[24] A. Fenyi, J. G. Davis, and K. Riverson, "Comparative Analysis of Advanced Encryption Standard , Blowfish and Rivest Cipher 4 Algorithms Abstract," *Int. J. Innov. Res. Dev*, vol. 3, no. 11, pp. 384–392, 2014.
[25] N. A. A. S. Bolaji and A. B. Abubakar, "Comparative Analysis of Encryption Algorithms," *Covenant J. Informatics Commun. Technol*, vol. 6, no. 1, pp. 16–30, 2018.
[26] S. K *Ang, "randomness_testsuite/README.md at master · stevenang/randomness_testsuite · GitHub*.