Performance Evaluation of Cryptographic Algorithms: DES, 3DES, Blowfish, Twofish, and Threefish

Haneen Alabdulrazzaq[1], Mohammed N. Alenezi[1]

[1]Computer Science & Information Systems Department , Public Authority for Applied Education & Training, Kuwait

***Abstract*:** Many individuals and organizations use the Internet to store and send personal or business information. Some of this information is highly confidential, and its online storage and transmission raises issues of data privacy and confidentiality. Major advances in Internet technology have aided intruders in obtaining unauthorized access to confidential information. The confidential information transmitted via the Internet must be protected and this can be achieved through cryptographic encryption and decryption algorithms. Encryption hides confidential information by converting it to an unreadable form. The reverse process of retrieving data from the unreadable or encrypted form is known as decryption. Many cryptographic algorithms exist today, and the selection of which one to use depends on several factors and measures. This paper presents a comparison of the encryption speeds of five different cryptographic symmetric block-cipher algorithms: DES, TripleDES, Blowfish, Twofish, and Threefish, based on the results of a simulations conducted with various text file sizes using Python. The results show that Blowfish outperforms the other algorithms tested.

***Keywords*:** Information security, Encryption, Decryption, Cryptography, Key, Cipher**.**

1. Introduction

Government entities, private companies, and individuals share data and information over the Internet. A significant amount of this data is private and must remain confidential between the exchanging parties. To ensure data confidentiality and security during transmission, Internet service providers rely heavily on cryptographic encryption and decryption algorithms.

Data encryption transforms regular text into an unreadable form. The inverse of encryption is decryption, in which encrypted data is reverted to its original form. The processes of encryption and decryption involve the use of certain keys. The main goal of encryption is to make decryption impossible to occur in the absence of those keys. There are many cryptographic algorithms in existence today. Most fall into three main categories: symmetric key algorithms, asymmetric key algorithms, and hashing algorithms.

Symmetric key algorithms use the same unique key to encrypt and decrypt data. This private (also referred to as secret) key is shared between the sender and the receiver using a secure communication medium. The data to be encrypted are handled either as blocks or as streams of ciphers. Symmetric key algorithms that use block ciphers divide data into blocks of fixed lengths, whereas stream cipher method encrypts data as a stream of bits.

Asymmetric key algorithms require the use of two keys, one of which is public and the other of which is private. The public key is used for data encryption, and the private key is used for decryption. Both keys are related and are derived mathematically. Asymmetric key algorithms have higher central processing unit (CPU) utilization requirements than symmetric key algorithms. They also require more time to complete the encryption and decryption operations, especially when dealing with large file sizes.

Hashing is a type of cryptographic algorithm that is used mainly for storing passwords and performing data integrity checks. Hash functions are one-way functions that map data into a fixed-size string of bits that is referred to as a hash. There are several different types of hash functions, including the secure hashing algorithm (SHA), RACE integrity primitives evaluation message digest (RIPEMD), message digest algorithm (MD), and Whirlpool, among others.

Digital signatures are mathematical functions or algorithms that can be used to ensure the authenticity of an email message, a credit card transaction, or a digital document. A digital signature can be thought of as being similar to an electronic fingerprint used to identify and protect users. A digital signature is used to ensure that a message or document has not changed since the time when its digital signature was signed. A digital signature is applied to a document by encrypting the document with the sender's secret key after hashing the document or information. Public key infrastructure (PKI), which is a set of standards and policies for the distribution of public keys and validation of the identity of users with digital certificates, is used to strengthen security.

We evaluated the performance of five different symmetric key cryptographic algorithms: DES, 3DES, Blowfish, Twofish, and Threefish. Each of these algorithms was tested for speed and throughput for different file sizes. The remainder of this paper is organized as follows. Section 2 briefly describes the five algorithms evaluated. Section 3 summarizes related work and experiments conducted to evaluate the performance of cryptographic algorithms. Section 4 presents the simulation setup and the results obtained. Section 5 presents conclusions drawn from the results and recommendations for future research.

2. Overview Of The Evaluated Cryptographic Algorithms

2.1 Data Encryption Standard (DES)

The National Bureau of Standards (NBS) accepted IBM's Lucifer cipher, with some modifications, as the Data Encryption Standard (DES) in 1973. The NBS adopted the standard as the Federal Information Processing Standard (FIPS) in 1977 [1]. DES is one of the earliest block cipher-based symmetric key cryptographic algorithms. It encrypts or decrypts blocks of 64-bit data using the same 56-bit key. The actual key length is 56 bits, and it also contains odd parity bits, making a total of 64 bits. Because of its small key size, DES was deemed insecure and has been replaced with the Advanced Encryption Standard (AES).

The DES algorithm encrypts or decrypts data in 16 Feistel rounds of operation and two permutations (P-boxes): an initial permutation and a final one. At each round, the algorithm uses different key combinations obtained from the original 56-bit key. There are 256 possible combinations for the given key. Figure 1 shows the working structure of the DES algorithm.

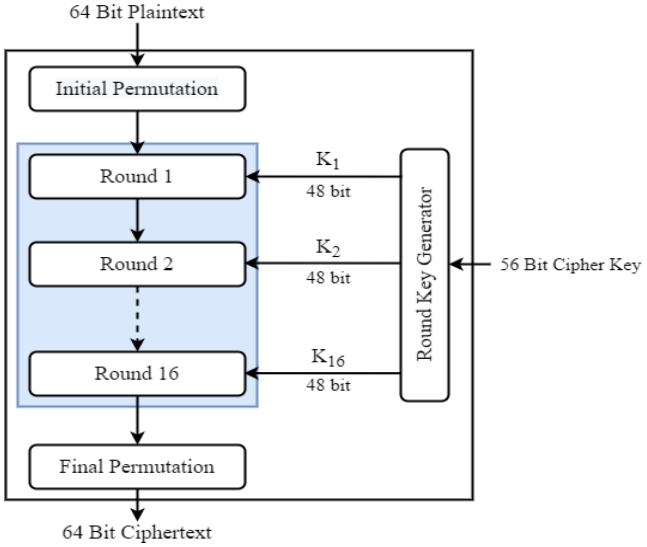

Figure 1. General structure of DES [2]

The initial and final permutations are keyless operations. In the initial permutation, the first bit of input is overwritten by the 58th bit, the second bit is overwritten by the 50th bit, and so on. Table I lists the permutation rules for 64-bit data. Notice that after the initial permutation, the first bit of input is in the 40th position, and this becomes the first position in the final permutation [3]. The two permutations are inverses of each other. The purpose these permutations serve has not been revealed by the algorithm's designers.

Table 1. Permutation Table

Initial Permutation	Final Permutation
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

The working of DES is based primarily on the substitution and transposition that are performed at each round. The resultant permuted block obtained after the initial permutation is divided into left and right sub-blocks, each 32 bits in size. Figure 2 illustrates the operations performed by DES.

The core of a DES-based algorithm is the DES function, which is shown in figure 3. The DES function takes the rightmost 32 bits of data (Rn) and 48-bit key as inputs and produces 32-bit output. The expansion D-box takes 32-bit input and converts it into 48 bits for XORing with a 48-bit key. The input data are first divided into eight subsections of four bits each, using a predetermined rule. The four bits are then expanded to six bits in such a way that the first bit is the fourth bit of the previous section, the next four bits are the same as the input four bits, and the last bit is the first bit of the next section. The resultant 48 bits are XORed with the key and then pass through the S-Boxes, which take 48-bit input and produce 32-bit output. Each S box takes six bits of data and outputs four bits of data.

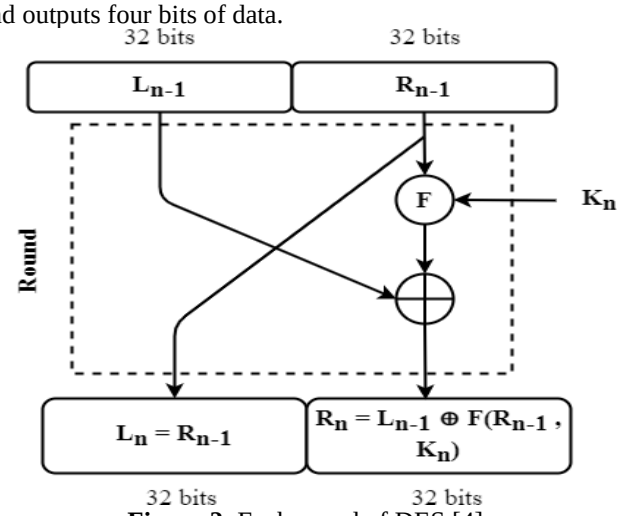

Figure 2. Each round of DES [4]

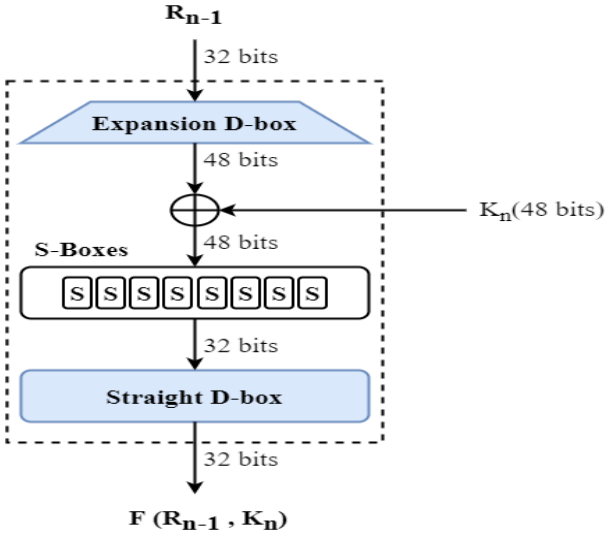

Figure 3. The working of DES F Function [5]

The round key generator generates sixteen distinct 48-bit keys for every 16 rounds from the 56-bit cipher key. The DES algorithm has several modes of operation, including the Electronic Code Book (ECB), Cipher Block Chaining (CBC), and Cipher Feedback (CFB) modes.

2.2 Triple Data Encryption Standard (3DES)

Triple DES, or 3DES as it is commonly known, was first published in 1998 [6]. It is also a block cipher-based symmetric algorithm, in which each 64-bit block of plaintext undergoes a DES cipher three times to enhance the security of the DES algorithm [7]. Separate 64-bit keys are used for each DES application. Triple DES enhances security but makes the encryption process three times slower than that of DES. The working of 3DES is illustrated in Figure 4.

3DES operates in four modes: DES-EEE3, DES-EDE3, DES-EEE2, and DES-EDE2. In DES-EEE3 and DES-EEE2, plain text undergoes DES encryption three times, using three different keys and two different keys, respectively. DES-EDE3 is performed in such a way that it undergoes DES encryption, then DES decryption, and finally DES encryption, using three keys. DES-EDE2 follows the same operation sequence as DES-EDE3 but uses only two keys.

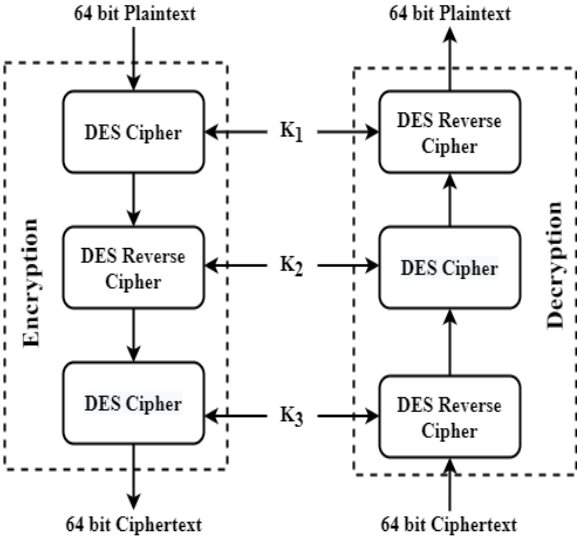

Figure 4. The working of 3DES [8]

2.3 Blowfish

Blowfish was designed and published in 1993 by Bruce Schneier. It is a good alternative to the existing encryption algorithms because it is an unpatented, freely available fast algorithm [6]. It is a symmetric block cipher-based algorithm that encrypts a block 64 bits in size. Blowfish uses variable-length keys varying in size from 32 bit to 448 bits. Blowfish is a Feistel cipher that encrypts data in 16 rounds of operations, as shown in Figure 5 [1]. Blowfish is considered secure and can be implemented easily.

The original plaintext (E) is divided into LE_0 and RE_0, each of which is 32 bits in size. Blowfish also has a P-array containing 18 subkeys, with each array element 32 bits in size. In each round, the inputs (LE and RE) passed to the next round are calculated using equations 1 and 2.

$$LE_n = LE_{n-1} \oplus EP_n \quad (1)$$

$$RE_n = Fn(LE_n) \oplus RE_{n-1} \quad (2)$$

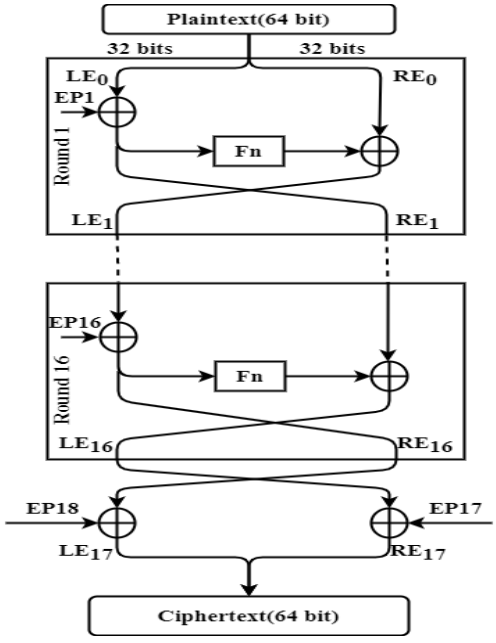

Figure 5. The working of Blowfish [1]

RE and LE are swapped before the next round is started. After 16 rounds of this repeated step, the final ciphertext can be formed by combining LE17 and RE17. Once the final round has been completed, LE16 and RE16 are again swapped to undo the effect of the last swapping. LE17 is calculated by XORing LE16 with EP18, and similarly, RE17 is calculated by XORing RE16 with EP17 [1] [3].

The Blowfish function Fn consists of four key-dependent substitution boxes (S-boxes), each of which can take 8-bit data as input and produce 32-bit data as output, as shown in Figure 6. Each time the Blowfish function is applied, it divides the LE into four subsets of 8-bit data, which are given to the corresponding S-Boxes. The output of each S-Box is taken and combined as shown in Figure 6.

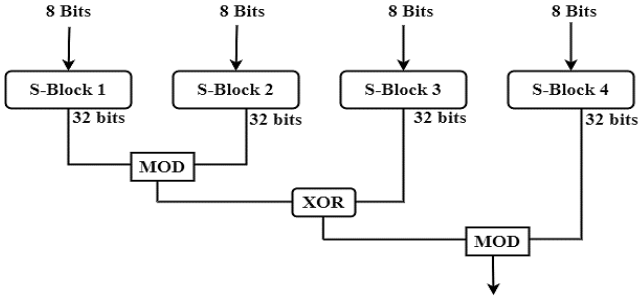

Figure 6. The Blowfish F Function [9, 10]

2.4 Twofish

Twofish is a highly flexible and secure 128-bit symmetric block cipher based cryptographic algorithm. It has a 16-round Feistel structure, as do DES and Blowfish. Twofish uses variable-length keys of sizes 128, 192, and 256 bits. Half of each key is worked as an actual cipher key, and the other half is used to modify the encryption algorithm [10]. The algorithm uses a bijective F function consisting of four key-dependent 8-by-8 S-boxes, a fixed 4-by-4 maximum distance separable (MDS) matrix, a pseudo-Hadamard transform (PHT), bitwise rotations, and a well-designed key scheduler [11], as shown in Figure 7. 128-bit plain text is split into four subsections of 32 bits each. These four subsets undergo an initial whitening phase with four subkeys. In the whitening step, the XOR operation is performed with the corresponding key.

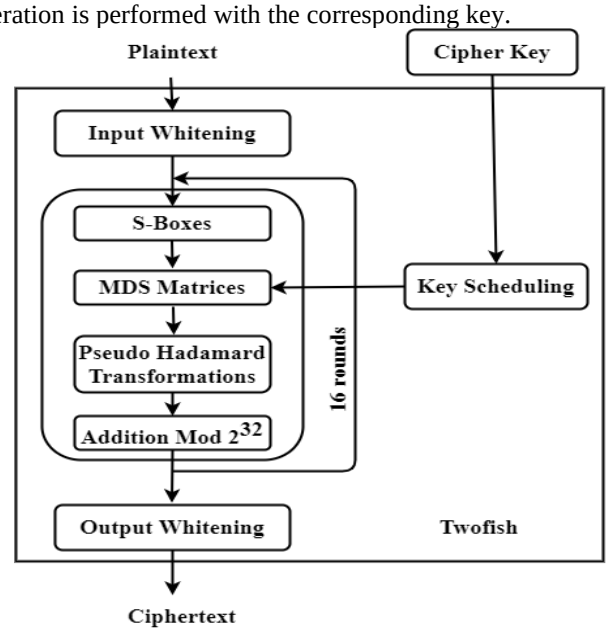

Figure 7. Block diagram of Twofish [11]

The core of Twofish is the g function, which contains the S-boxes and MDS matrix. The actual working of each round of Twofish is shown in Figure 8. Each input of its g function is further divided into four, each subset runs through its S-box, and the output is combined in an MDS matrix. The resulting 32-bit matrix is the output of the g function. The output of both g functions is further combined using PHT.

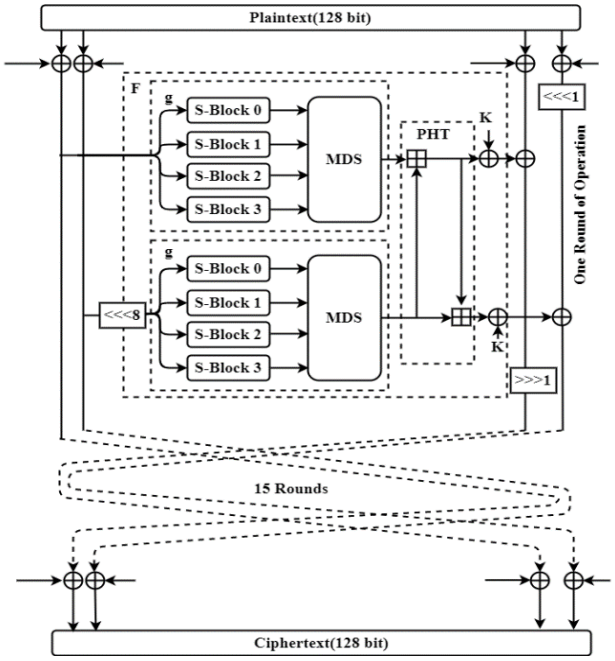

Figure 8. The Working of Twofish [10]

Twofish can run easily on smaller processors, such as smart cards, and can be embedded in hardware. It is one of the license-free and unpatented encryption mechanisms available, and it can be customized [12].

2.5 Threefish

Threefish is a tweakable symmetric block cipher-based cryptographic algorithm that takes an additional tweak value of 128 bits for all block sizes along with the plaintext and actual key value. This unique tweak value is used to encrypt the data. The key and block sizes in Threefish are equal. Threefish can encrypt data blocks that are 256 bits, 512 bits, or 1024 bits in size, using a key of equal size. It typically takes 72 rounds to perform encryption for data blocks that are 256 or 512 bits in size. In the case of a 1024-bit block, however, it takes 80 rounds of operation to produce ciphertext. To avoid timing attacks, Threefish does not utilize S-boxes or any other table lookups [13]. Threefish encryption uses three types of operations: addition, XOR, and rotations.

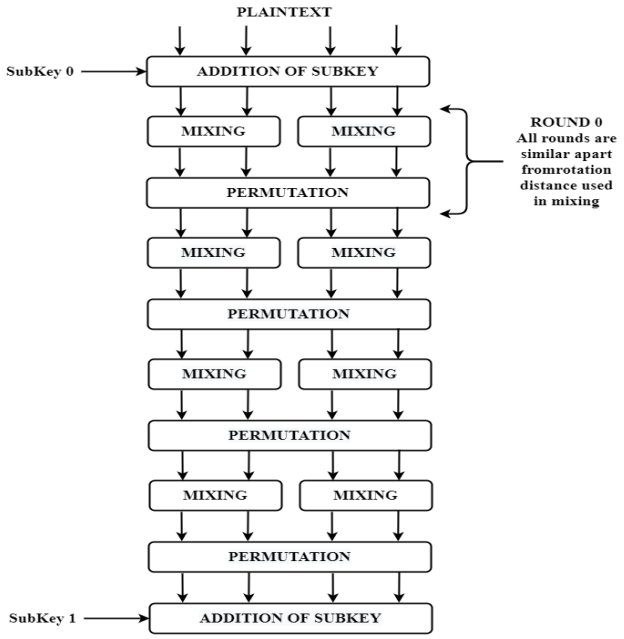

Figure 9. First Four Round Operations of the Threefish-256 Cipher [14]

Figure 9 illustrates the first four rounds of a Threefish-256 cipher. One subkey is generated after every four rounds. At each round, the word permutation is the same, and rotations are constant in the consecutive eight rounds. The key schedule consists of a key and a tweak value, and it generates subkey values [14]. Threefish-256 consists of 72 rounds with two mix operations followed by a permutation in each round. Threefish-512 consists of 72 rounds with four mix operations in each round, and Threefish-1024 has 80 rounds with eight mix operations in each round. Threefish is considered a wide-block cipher algorithm because it works on blocks larger than 128 bits.

2.6 Modes of Block Cipher based encryption

Symmetric key encryption uses the same key for encrypting and decrypting data blocks. Multiple blocks encrypted using the same key can weaken the encryption process, and intruders can hack a message if it contains similar blocks of data. To handle this issue of generating identical ciphertext from identical plaintext, extra input is introduced to each block of encryption. This idea of adding plaintext and ciphertext from the previous block is known as a block cipher mode of operation. Several block cipher modes exist today to enhance the security of symmetric key cryptographic algorithms, including the Electronic Code Book (ECB), Cipher Block Chaining (CBC), Propagating or Plaintext Cipher Block Chaining (PCBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). Each of these modes has its own advantages and disadvantages, as summarized below.

2.6.1 Electronic Code Book (ECB)

ECB is a straightforward mode in which each block is encrypted separately. The plaintext is divided into n-bit blocks, and each block is encrypted in any order. Decryption can also be performed individually without consideration of the order. The main advantages of ECB mode are that it is fast and easy to implement, no data synchronization is needed; encryption and decryption can be done in parallel, and single-bit errors only affect the corresponding block. However, ECB can be easily deciphered and subjected to substitution attacks [15].

2.6.2 Cipher Block Chaining (CBC)

CBC mode introduces a degree of randomness to avoid attacks due to determinism. The plaintext is split into n-bit blocks, and each block is XORed with the ciphertext block of the previous block, except for the first block. The first block is XORed with a random initialization vector (IV) of the same length as the plaintext. IV is a number used only once (nonce) that can be generated using a random number generator or counter. The retrieval of plaintext from ciphertext is almost impossible in the case of a single-bit error in plaintext. In contrast, a single-bit error in the ciphertext would affect only two subsequent plaintext blocks. Encryption and decryption cannot be done in parallel because the ciphertext of each block affects the next block's encryption.

2.6.3 Plaintext Cipher Block Chaining (PCBC)

PCBC mode is similar to CBC mode in that it combines the plaintext and ciphertext current block with the plaintext of the next block. In addition, a single-bit transmission error will damage an entire block of data, preventing retrieval of the plaintext.

2.6.4 Cipher Feedback (CFB)

CFB mode works in a manner similar to a stream cipher, in which the plaintext of the current block is combined with the ciphertext of the previous block. In CFB mode, the same encryption procedure is used for both encryption and decryption. An initialization vector is XORed with the plaintext of the first block to produce the ciphertext. Parallelization is possible for decryption but not encryption. In this mode as well, a single-bit error in the plaintext will damage the entire ciphertext, but similar errors in the ciphertext will only affect the two subsequent blocks.

2.6.5 Output Feedback (OFB)

The working of OFB is similar to CFB. The n-bit plaintext in each block is XORed with the previously generated ones from the block cipher, except for the first block. The first block is XORed with the externally supplied initialization vector (IV). Encryption and decryption proceed in the same manner. The effect of one XOR nullifies another XOR. The feedback to each block can be performed before the actual plaintext. Parallelization of encryption and decryption is not possible. A single-bit error affects only the corresponding plain or ciphertext.

2.6.6 Counter (CTR)

The counter mode, like a stream cipher, encrypts data using an additional input that is a combination of an increasing counter and a nonce value. The nonce initial vector's length is less than the block length, and the counter is normally initialized as 0. This mode has been widely adopted because of its effectiveness. In counter mode, encryption and decryption are performed in parallel.

2.6.7 Encrypt-Authenticate-Translate (EAX)

EAX is a mode proposed by Mihir Bellare, Phillip Rogaway, and David Wagner for solving the problem of authenticated encryption with associated data (AEAD) [16]. EAX follows a two-pass scheme in which encryption and authentication are done independently of one another. In EAX mode, only the encryption functionality of the block cipher is used [17].

3. Related Work

Security becomes a primary criterion in the development of information technology and the Internet. Users need to keep their information safe and secure in storage and/or transmission. Service providers ensure security through data encryption and decryption. There are a variety of encryption and decryption algorithms available to protect the data. Users and service providers need to select encryption algorithms based on their needs. Each algorithm has its advantages and disadvantages. Our primary concern is choosing a suitable algorithm for a particular situation. Various studies have been done to compare the performance of encryption algorithms based on multiple parameters.

Bhanot and Hans [13] evaluated the performance of various symmetric and asymmetric cryptographic algorithms to identify the best algorithm among them. The authors analyzed the strengths and weaknesses of ten algorithms based on parameters such as the development, key length, number of rounds needed for encryption and decryption, block size, various types of attacks found, level of security, and encryption speed. The strength of each algorithm was found to depend on the parameters chosen and the situation. The authors shortlisted Blowfish and ECC for their speed and security. Among these algorithms, Blowfish had not been broken yet, whereas ECC had successfully been broken.

Wahid et al. [18] analyzed the performance of the DES, 3DES, AES, RSA, and Blowfish encryption algorithms to assess their performance, strengths, and weaknesses. They analyzed these algorithms based on various parameters, such as memory, time, and attacks. Blowfish outperformed the other algorithms in terms of memory, time, and level of security. AES was found to be the best algorithm in terms of confidentiality and integrity.

Tyagi and Ganpati [19] performed a theoretical analysis of the most popular cryptographic algorithms: DES, 3DES, AES, and Blowfish. They analyzed the performance of these block cipher-based symmetric key cryptographic algorithms with respect to various parameters, such as speed, block size, security against attacks, confidentiality, throughput, power consumption, and key size, among others. Blowfish outperformed the other three algorithms in terms of encryption or decryption time and throughput, whereas 3DES exhibited the poorest performance. Princy [20] assessed the performance of AES, DES, 3DES, Blowfish, RC4, and RC6 in terms of processing time and required number of rounds. Blowfish was found to provide higher security and privacy, even for an unsecured transmission channel. The results of the study also showed that the effectiveness of Blowfish could be increased by increasing the key length from 128 bits to 448 bits.

Mathur and Kesarwani [21] evaluated the performance of six widely used cryptographic algorithms(DES, 3DES, AES, RC2, RC6, and Blowfish) with respect to various parameters, including key length, encoding method, data type, and packet size. They evaluated the performance of these algorithms using both hexadecimal base encoding and base-64 encoding. The study results confirmed that the encoding technique has no impact on the performance. Blowfish performed better than all of the other algorithms under study in several respects. They performance of the algorithms in image encryption was also evaluated. RC2, RC6, and Blowfish were found to have significant difficulties in handling image encryption. The time and power consumption were found to increase with increasing key length. Nema and Rizvi [22] studied the performance of several available cryptographic algorithms (DES, 3DES, AES, Blowfish, Twofish, Threefish, RC2, RC4, RC5, and RC6) with respect to factors such as throughput, scalability, security, memory usage, power consumption, speed, and flexibility. Each of the algorithms was found to have advantages and disadvantages, depending on the purpose of the encryption and the parameters under consideration. Based on the study results, the researchers recommended that a user select the algorithm best suited for the application and the user's concerns. If the user's concerns are security, flexibility, memory usage, and encryption performance, then Blowfish is the best choice.

Nadeem and Javed [23] compared the performance, including the encryption speed, of the DES, 3DES, AES, and Blowfish algorithms for various input file contents and lengths and different hardware platforms. These block cipher-based symmetric algorithms were implemented using Java and were ranked in the following order of encryption speed: Blowfish, DES, AES, and Triple DES. Blowfish performed better than the other three algorithms in several respects. The encryption speeds of these algorithms increased with decreasing key length and increasing data block length, and while the security of each algorithm increased with increasing number of rounds, the encryption speed decreased. Alenezi et al. [24] compared

and analyzed AES, Blowfish, DES, DESede, SEED, IDEA, RC2, RC4, RC6, SEED, and XTEA for encryption time, throughput, and CPU utilization. The results showed that AES was the better candidate in terms of performance as well as the level of security it provided.

Jeevalatha and SenthilMurugan [10] analyzed various encryption algorithms, such as AES, Blowfish, and Twofish, to identify the algorithm that provided the highest security using the least space and time. They examined various aspects of the design and performance of each algorithm. Their results showed that AES performed better than the other two algorithms. Gehlot [11] proposed a modification of the existing Twofish encryption algorithm. The suggested minimum-delay Twofish algorithm achieved better performance.

4. Performance and Analysis

4.1 Simulation and System Setup

A performance simulation of the algorithms described in section 2 was performed using the Python programming language. The Crypto.Cipher package from Pycryptodome.org was used. This is a self-contained cryptographic library for Python that includes implementations of the DES, 3DES, and Blowfish cryptographic algorithms [25]. For the Twofish algorithm, an implementation found in Python called twofish was used [26]. The Threefish cryptographic algorithm's implementation was carried out using the Python package Pyskein-1.0 [27].

The encryption speeds of the five cryptographic algorithms were tested using generated text files 1 Mb, 5 MB, 10 Mb, 50 MB, and 100 MB in size. For the DES algorithm, the key size used in the simulation was 64 bits. For the 3DES, Blowfish, and Twofish algorithms, the key size used in the simulation was 128 bits. For the Threefish algorithm, the key size was 256 bits , and the tweak size was 128 bits. The EAX mode was used in the simulations with the DES, 3DES, and Blowfish algorithms. No operating mode was used in the simulations with the Twofish and Threefish algorithms.

The simulations were executed on a computer with a 64-bit Microsoft Windows 10 Pro operating system. The computer's CPU has two Intel™ processors running at 2.927 GHz and 64 GB of RAM. The system has Python version 3.10 installed.

4.2 Results and Discussion

The simulation results for the five algorithms are shown in Figures 10 through 14. Figure 15 shows a comparison of the performance of the algorithms tested. Table 2 lists the recorded speed, measured in seconds, for all of the algorithms. In the case of the 1-MB file, the DES algorithm's encryption speed was 0.042 s, second only to that of Blowfish. Figure 10 shows the encryption speed for the DES algorithm. The 3DES algorithm's encryption speed was 0.139 s for a file size of 1 MB and 11.973 s for 100 MB, as illustrated in Figure 11. The results for 3DES were well within expectations, in that its encryption speeds were approximately three times faster than the DES algorithm. Blowfish proved to have the fastest encryption speeds, encrypts a 100-MB text file in 3.193 s, for example. The encryption times for Blowfish are shown in Figure 12. The encryption times for Twofish included 0.514 s for a 1-MB file and 51.886 s for a 100-MB file. According to the simulation results, Twofish was the slowest of the five algorithms tested, as shown in Figure 13. The Threefish algorithm achieved relatively fast encryption speeds, given the complexity of the algorithm and the key size used. Figure 14 illustrates the encryption speeds for the Threefish algorithm.

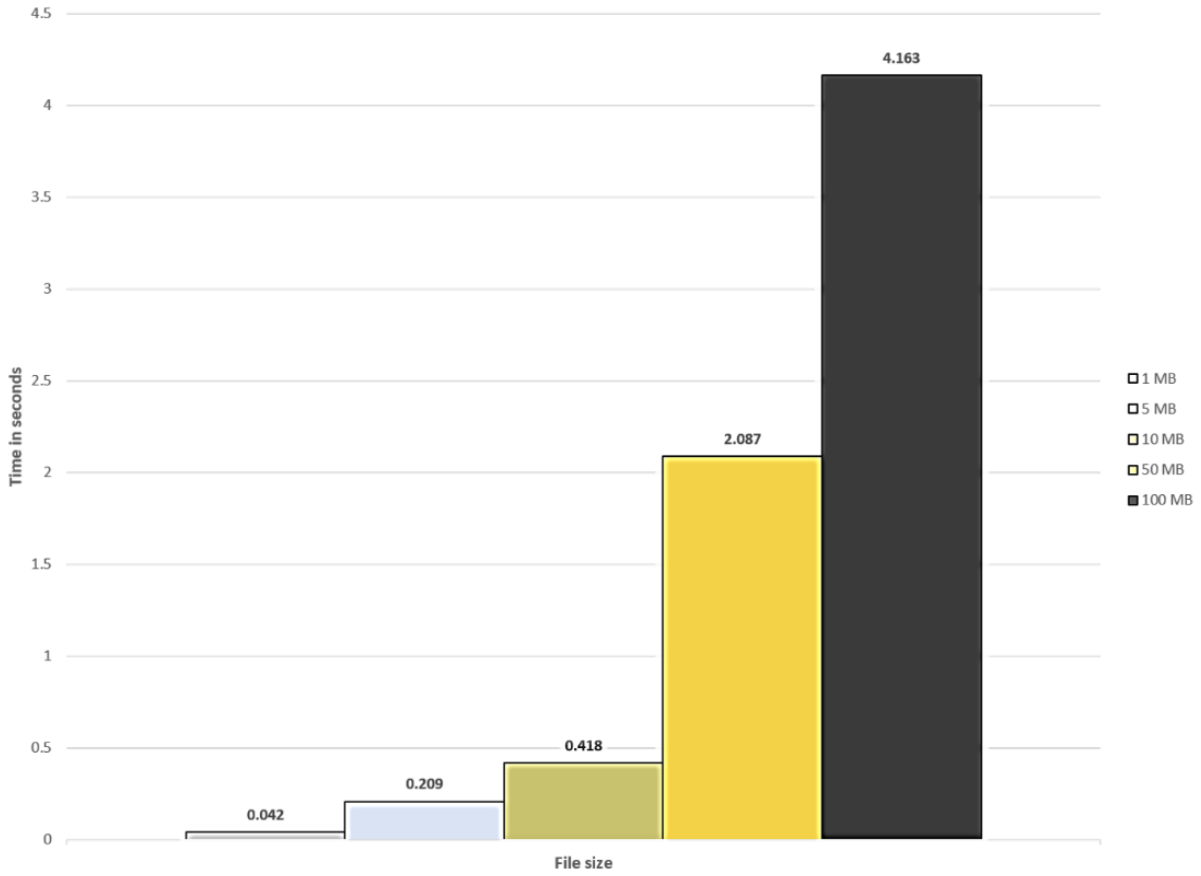

Figure 10. Encryption speed for DES algorithm

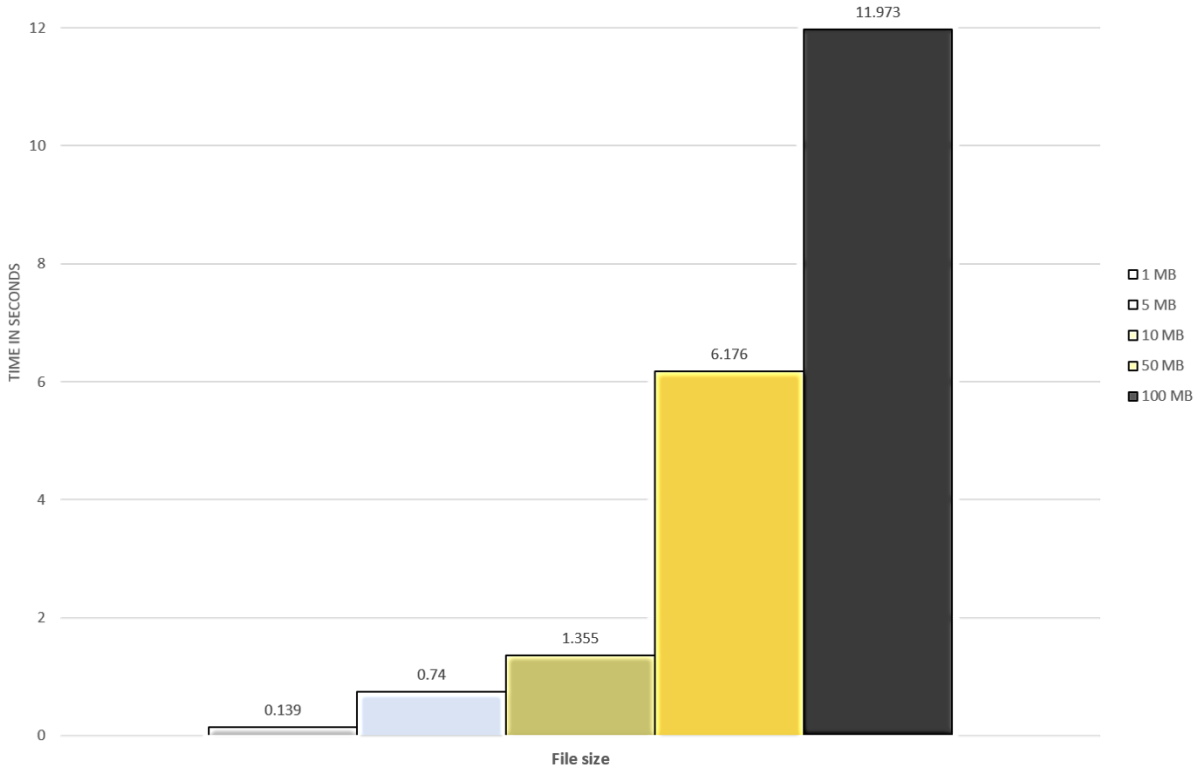

Figure 11. Encryption speed for 3DES algorithm

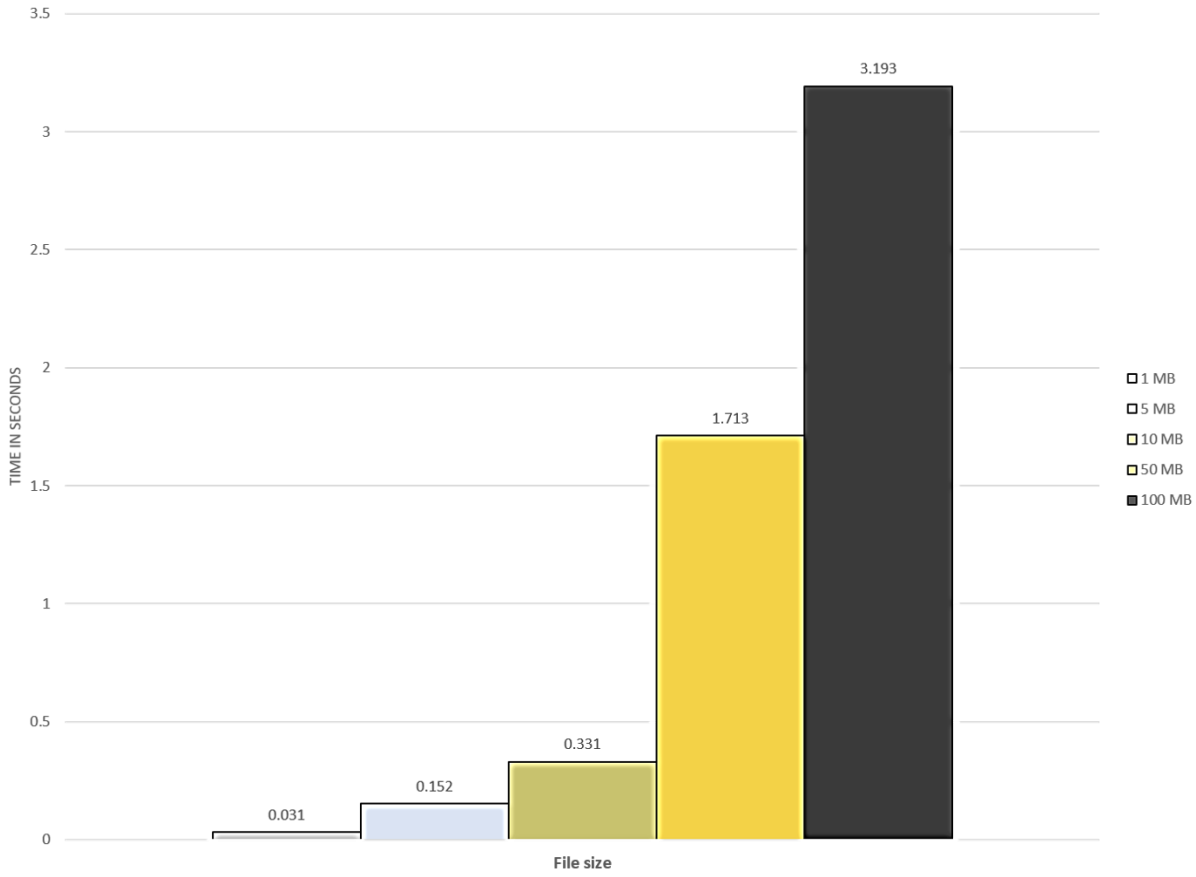

Figure 12. Encryption speed for Blowfish algorithm

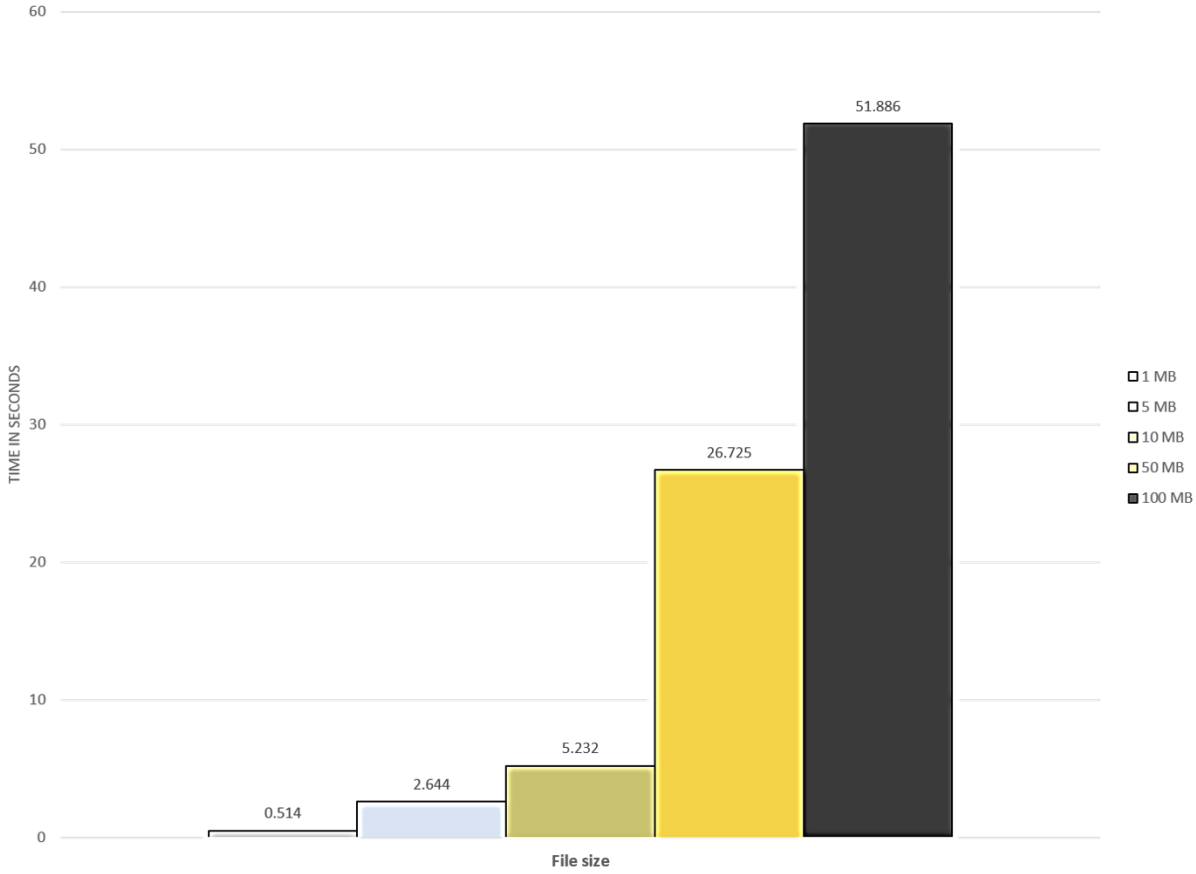

Figure 13. Encryption speed for Twofish algorithm

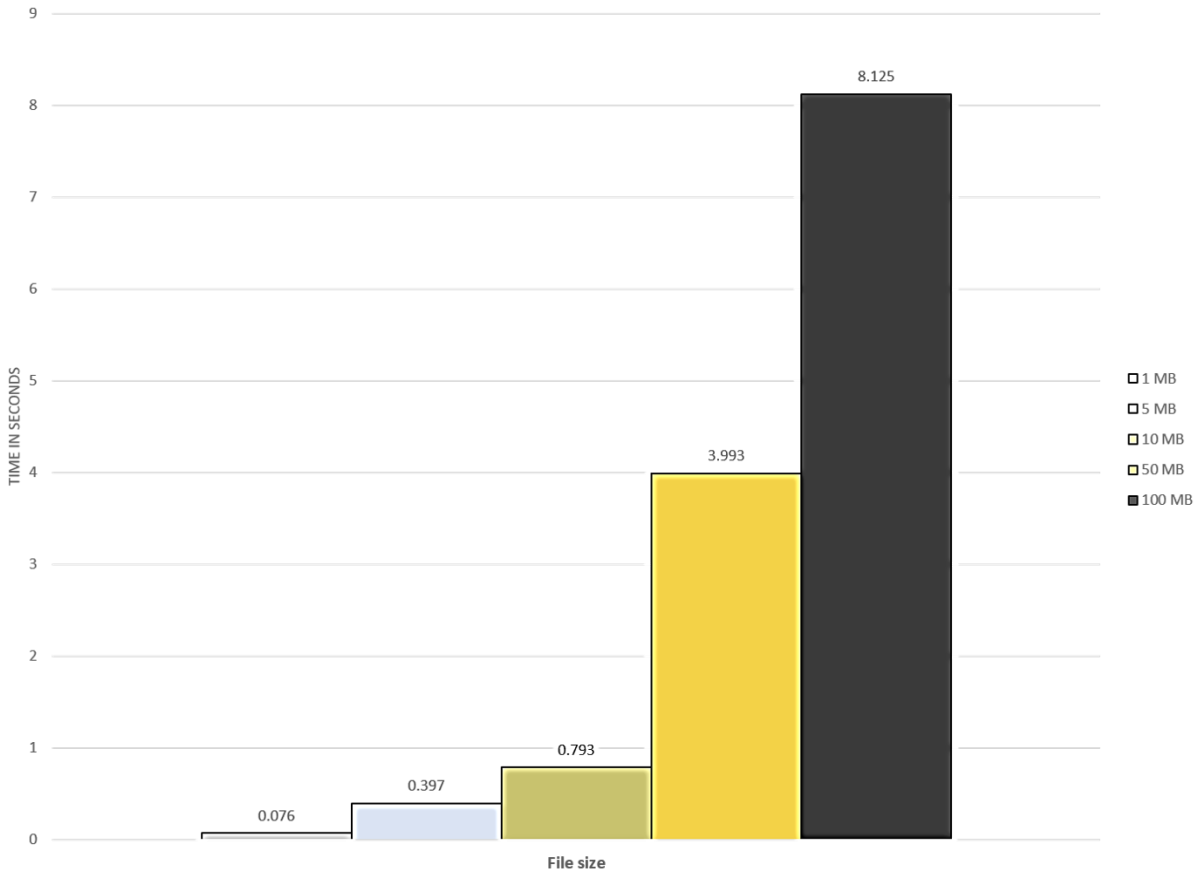

Figure 14. Encryption speed for Threefish algorithm

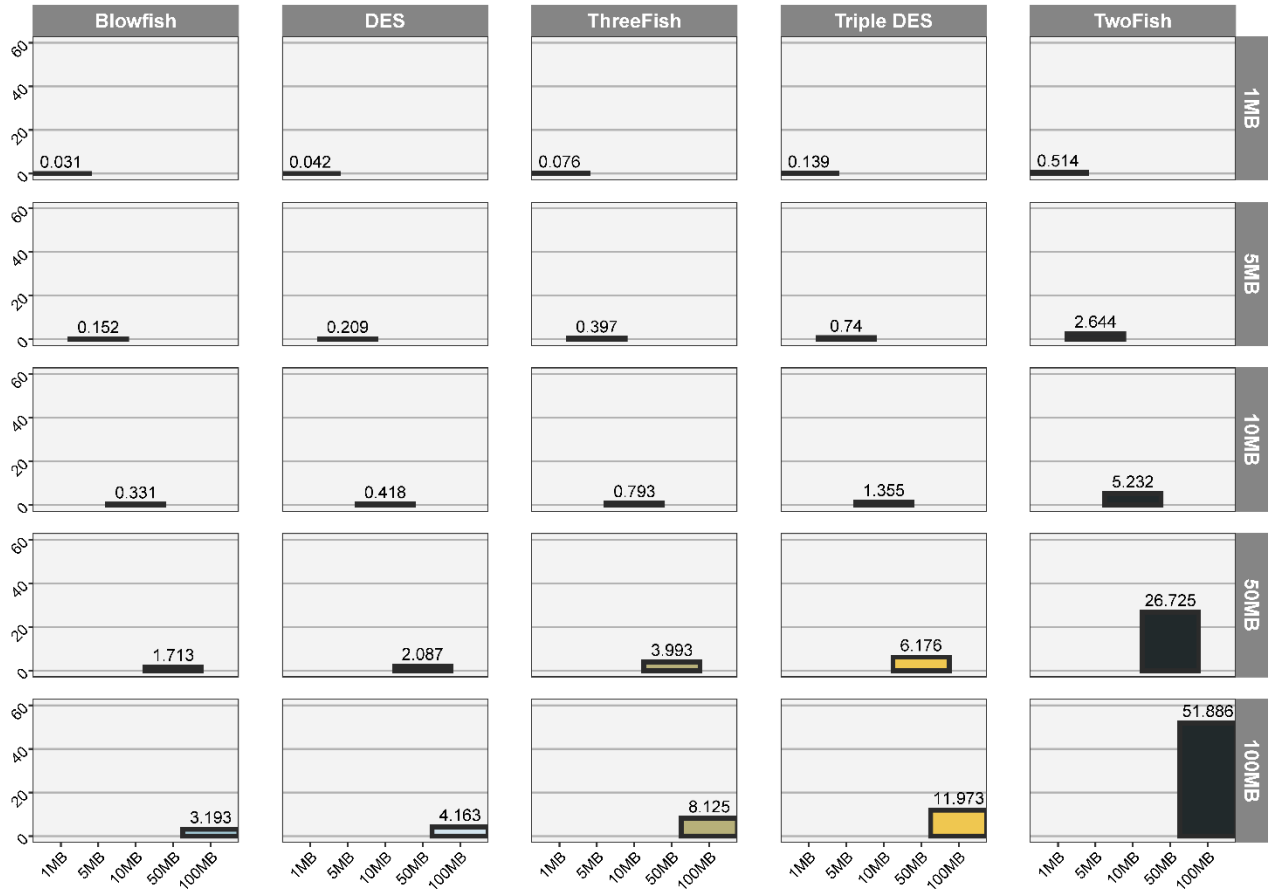

Figure 15. Comparison of encryption speed for DES, 3DES, Blowfish, Twofish, and Threefish.

Table 2. Comparison of Encryption Time Measured in Seconds

Algorithm	Key size in bits	1 MB	5 MB	10 MB	50 MB	100 MB
DES	64	*0.042*	*0.209*	*0.418*	*2.087*	*4.163*
3DES	128	*0.139*	*0.74*	*1.355*	*6.176*	*11.973*
Blowfish	128	*0.031*	*0.152*	*0.331*	*1.713*	*3.193*
Twofish	128	*0.514*	*2.644*	*5.232*	*26.725*	*51.886*
Threefish	256	*0.076*	*0.397*	*0.793*	*3.993*	*8.125*

The results show that the Blowfish algorithm is the fastest in terms of encryption. It should be noted that these results pertain to the particular experimental setup and the algorithm packages as described in IV-A. DES, 3DES, and Blowfish were tested in EAX operating mode. Applying other modes of operation may yield slightly different results. Twofish and Threefish are tested without a mode of operation. For real-world applications, an operating mode such as CBC or CTR is highly recommended by the twofish Python library author. With regard to Threefish algorithm, not all modes of operation may work, as many of the standard modes offer incomplete support for wide-block cipher algorithms [28]. Therefore, choosing an appropriate mode of operation for the Twofish and Threefish algorithms, as well as the rest of the algorithms, is left to the discretion of the user, depending on the requirements of the application.

5. Conclusion and Future Work

In this research, we compared the encryption speeds of five symmetric key algorithms: DES, 3DES, Blowfish, Twofish, and Threefish. The results show that Blowfish outperforms the other algorithms tested. These results should be interpreted in light of the particular experimental setup used and nature of the implementations, as described earlier. Potential users of cryptographic algorithms should consider several factors—not just encryption speed—before deciding on an algorithm to employ. Factors such as the type and size of file, the system in which the cryptographic algorithm will be executed, and the level of security needed must all be taken into consideration before a decision is made. Several algorithms, including DES, have been proven to be vulnerable to attacks and therefore might not be suitable for use when highly confidential data is at stake.

In future work, we would like to investigate the algorithms we tested with respect to several performance measures other than encryption speed. We would also like to test the performance of various encryption algorithms using different file types, such as images, audio, and video files. Finally, we would like to explore the effects of particular implementations of cryptographic algorithms on encryption speed, as well as on other performance measures.

References

[1] M. Mushtaq, S. Jamel, A. Disina, Z. Pindar, N. Shakir, and M. Mat Deris, "A survey on the cryptographic encryption algorithms," International Journal of Advanced Computer Science and Applications, Vol. 8, No. 11, pp. 333–344, 2017.

[2] A. Shorman and M. Qatawneh, "Performance improvement of double data encryption standard algorithm using parallel computation," International Journal of Computer Applications, vol. 179, No. 25, pp. 1–6, 2018.

[3] A. Kahate, Cryptography and Network Security. Tata Mcgraw-Hill Publishing Company Limited, 2003. [Online]. Available: https://books.google.com.kw/books?id=SWbn3lBe2FcC

[4] H. Zodpe and P. Wani, "Design and implementation of algorithm for des cryptanalysis," International Conference on Hybrid Intelligent Systems (HIS), pp. 278–282, 2012, India.

[5] B. J. Saha and K. Kabi, "Digital image encryption using ecc and des with chaotic key generator," International Journal of Engineering Research & Technology (IJERT), Vol. 2, No. 11, pp. 2593–2597, 2013.

[6] P. Patil, P. Narayankar, D. Narayan, and S. M. Meena, "A comprehensive evaluation of cryptographic algorithms: Des, 3des, aes, rsa and blowfish," Procedia Computer Science, Vol. 78, pp. 617–624, 2016.

[7] M. Panhwar, S. Ali Khuhro, G. Panhwar, and K. Ali, "Saca: A study of symmetric and asymmetric cryptographic algorithms," IJCSNS International Journal of Computer Science and Network Security, Vol. 19, No. 1, pp. 48-55, 2019.

[8] C. Rahmad, K. Arai, A. Prasetyo, and N. Arizki, "Noble method for data hiding using steganography discrete wavelet transformation and cryptography triple data encryption standard: Des," International Journal of Advanced Computer Science and Applications, Vol. 9, No. 11, pp. 261-266, 2018.

[9] R. Patel and P. Kamboj, "Security enhancement of blowfish block cipher," International conference on smart trends for information technology and computer communications, pp. 231–238, 2016, India.

[10] E.Jeevalatha and S.SenthilMurugan, "Evolution of aes, blowfish and two fish encryption algorithm," International Journal of Scientific and Engineering Research, Vol. 9, No. 4, pp. 115-118, 2018.

[11] P. Gehlot, S. R. Biradar, and B. P. Singh, "Implementation of modified twofish algorithm using 128 and 192-bit keys on vhdl," International Journal of Computer Applications, Vol. 70, No. 13, pp. 37–42, 2013.

[12] B. Schneier, "The twofish encryption algorithm," https://www.schneier.com/academic/archives/1998/12/the_twofish_encrypti.html, Dec. 1998, [Online; accessed 11 Nov 2021].

[13] R. Bhanot and R. Hans, "A review and comparative analysis of various encryption algorithms," International Journal of Security and Its Applications, Vol. 9, No. 4, pp. 289-306, 2015.

[14] N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Callas, and J. Walker. "The Skein hash function family." Submission to NIST (round 3) 7, no. 7.5, 2010.

[15] D. Bujari and E. Aribas, "Comparative analysis of block cipher modes of operation," International Advanced Researches & Engineering Congress, pp. 1-4, 2017, Turkey.

[16] M. Bellare, P. Rogaway, and D. Wagner, "The eax mode of operation," International Workshop on Fast Software Encryption. Springer, pp. 389–407, 2004, Berlin, Heidelberg.

[17] E. B. Kavun, H. Mihajloska, and T. Yalcin, "A survey on authenticated encryption–asic designer's perspective," ACM Computing Surveys (CSUR), Vol. 50, No. 6, pp. 1-21, 2019.

[18] M. N. A. Wahid, A. K. Ali, B. Esparham, and M. Marwan, "A comparison of cryptographic algorithms: Des, 3des, aes, rsa and blowfish for guessing attacks prevention," Journal Computer Science Applications and Information Technology, Vol. 3, No. 2, pp. 1-7, 2018.

[19] N. Tyagi and A. Ganpati, "Comparative analysis of symmetric key encryption algorithms," International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, No. 8, pp. 348-354, 2014.

[20] P. Princy, "A comparison of symmetric key algorithms des, aes, blowfish, rc4, rc6: A survey," International Journal of Computer Science & Engineering Technology (IJCSET), Vol. 6, No. 5, pp. 328-331, 2015.

[21] M. Mathur and A. Kesarwani, "Comparison between des, 3des, rc2, rc6, blowfish and aes," Proceedings of National Conference on New Horizons in IT-NCNHIT, Vol. 3, pp. 143-148, 2013, India.

[22] P. Nema and M.A.Rizvi, "Critical analysis of various symmetric key cryptographic algorithms," International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 3, No. 6, pp. 4301-4306, 2015.

[23] A. Nadeem and M. Y. Javed, "A performance comparison of data encryption algorithms," 2005 international Conference on information and communication technologies. IEEE, pp. 84–89, 2005, Pakistan.

[24] M. N. Alenezi, H. Alabdulrazzaq, and N. Q. Mohammed, "Symmetric encryption algorithms: review and evaluation study," International Journal of Communication Networks and Information Security, Vol. 12, No. 2, pp. 256-272, 2020.

[25] L. GitHub, "Crypto.cipher package," https://pycryptodome.readthedocs.io/en/latest/src/cipher/cipher.html, Jun. 2014, [On- line; accessed 12 Dec 2021].

[26] N. Ferguson, "twofish 0.3.0," https://pypi.org/project/twofish/, Nov. 2013, [Online; accessed 12 Dec 2021].

[27] H. Fu¨rstenau, "Pyskein 1.0 - the skein hash algorithm for python," https://pythonhosted.org/pyskein/index.html, Aug. 2013, [Online; accessed 12 Dec 2021].

[28] C. wiki, "Threefish," https://www.cryptopp.com/wiki/Threefish, Sep. 2021, [Online; accessed 14 Dec 2021].