DOI: 10.24996/ijs.2023.64.5.40

ISSN: 0067-2904

Fast 128-bit Multi-Pass Stream Ciphering Method

Safa S. Abdul-Jabbar [1]*, Abeer E. Abed[2], Sajaa G. Mohammed[3], Faisel G. Mohammed[4]

[1] Department of Computer Science, College of Science for Women, University of Baghdad, Baghdad, Iraq
[2] Office of the Vice President for Scientific Affairs, University of Baghdad, Baghdad, Iraq
[3] Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq
[4] Department of Remote Sensing & GIS, College of Science, University of Baghdad, Baghdad, Iraq

Received: 22/9/2022 Accepted: 18/11/2022 Published: 30/5/2023

Abstract

Information is an essential and valuable object in all systems. The more information you have about your issue, the better you can conform to the world around you. Moreover, information recognizes companies and provides influence that helps one company be more effective than another. So, protecting this information using better security controls and providing a high level of access to authorized parties becomes an urgent need. As a result, many algorithms and encryption techniques have been developed to provide a high level of protection for system information. Therefore, this paper presents an enhancement to the Blowfish algorithm as one of the cryptography techniques. Then it proposes an enhancement for increasing efficiency and secrecy for this algorithm, which are the main criteria for these modifications. In this paper, the main modification made for the Blowfish algorithm is altering the S-box according to the principles of the block cipher (OFB). The results were approved using hamming distance and avalanche effect. The proposed algorithm presents advantages on several points, including reducing the required time for the encryption and decryption processes and preventing transmission errors from perpetuating all data. Also, we can conclude that the modified Blowfish algorithm can be used for all text encryption systems because of its flexibility (unlimited input text size and expanding key size).

Keywords: Multi-pass Algorithms, Blowfish, Cryptography, Block Cipher, Encryption Algorithms

طريقة تشفير دفق سريعة متعددة التمريرات 128 بت

صفا سامي عبد الجبار[1]*, عبير عيسى عبد[2], سجا غازي محمد[3], فيصل غازي محمد[4]

[1]قسم علوم الحاسوب، كلية العلوم للبنات، جامعة بغداد، بغداد، العراق
[2]مكتب مساعد رئيس الجامعة للشؤون العلمية، جامعة بغداد، بغداد، العراق
[3]قسم علوم الرياضيات، كلية العلوم، جامعة بغداد، بغداد، العراق
[4]قسم التحسس النائي ونظم المعلومات، كلية العلوم، جامعة بغداد، بغداد العراق

*Email: safa.s@csw.uobaghdad.edu.iq

الخلاصة

المعلومات هي شيء أساسي وقيِّم في جميع الأنظمة. كلما زادت المعلومات المتوفرة لديك حول مشكلتك، كان بإمكانك التوافق بشكل أفضل مع العالم من حولك. علاوة على ذلك، تعرف المعلومات بالشركات وتوفر التأثير الذي يساعد شركة ما على أن تكون أكثر فعالية من الأخرى. لذلك، فإن حماية هذه المعلومات باستعمال ضوابط أمنية أفضل وتوفير مستوى عالٍ من الوصول إلى الأطراف المصرح لها تصبح حاجة ملحة. نتيجة لذلك، تم تطوير العديد من الخوارزميات وتقنيات التشفير لتوفير مستوى عالٍ من الحماية لمعلومات النظام. لذلك، يقدم هذا البحث خوارزمية السمكة المنتفخة كأحد تقنيات التشفير. ثم يقترح تعزيزًا لزيادة الكفاءة والسرية لهذه الخوارزمية، وهي المعايير الرئيسية لهذه التعديلات. في هذا البحث، التعديل الرئيسي الذي تم إجراؤه لخوارزمية السمكة المنتفخة هو تغيير S–box بمبادئ تشفير الكتلة .(OFB) دققت النتائج باستعمال مسافة الطرق وتأثير الانهيار الجليدي. تقدم الخوارزمية المقترحة مزايا في عدة نقاط، بما في ذلك تقليل الوقت المطلوب لعمليات التشفير وفك التشفير ومنع أخطاء الإرسال من إدامة جميع البيانات. أيضًا، يمكننا أن نستنتج أنه يمكن استعمال خوارزمية السمكة المنتفخة المعدلة لجميع أنظمة تشفير النص نظرًا لمرونتها (حجم نص الإدخال غير المحدود وتوسيع حجم المفتاح).

1. Introduction

The widespread use of cryptography is a vital aspect of the information revolution. Cryptography is concerned with data security and confidentiality [1]. Since electronic communications on computer networks have appeared, conversations and transactions between people must be confident. Cryptography is one solution for this problem. Cryptography is a study that deals with encryption and decryption. Encryption converts data (i.e., plaintext) using a key into a form that cannot be read except by the authorized person (i.e., ciphertext). Decryption is the process of retrieving data from an authorized person. The result of encryption is known as a cipher. There are two types of ciphers: stream and block ciphers. In stream ciphers, the message units are bits, and a random bit generator usually produces the key. The plaintext is encrypted on a bit-by-bit basis [2, 3]. Furthermore, ciphers are classified into two types based on their symmetric key and asymmetric nature. In symmetric, the same key is used for encrypting plaintext and decrypting. In asymmetric encryption, one key is used to encrypt plaintext (i.e., the public key), and another key is used to decrypt cipher (i.e., the private key) [4].

Many algorithms are now utilized for data encryption and decryption. Broadly, these algorithms are classified into two categories based on the key used between the transmitter and receiver of the information: symmetric-key Cryptography uses a single key for encryption and decryption. Hence, it requires less computing time and reduces processing overhead [5].

Symmetric key cryptography can be employed in block ciphers and stream ciphers. The block cipher mode of operation works by taking the complete message as a single block. A stream cipher works by dividing the data into single bits. The separated bits are then randomized and used for encryption [6]. While asymmetric key cryptography necessitates the use of two distinct keys, one private and the other public [7].

Asymmetric cryptography is more popular than the other type. Because of the following differences between these two types [8]:

- Symmetric encryption encrypts and decrypts messages using a single key shared among a group of users who receive messages. In contrast, asymmetric encryption encrypts and decrypts messages using a pair of keys (public key and private key).

- Asymmetric was introduced to tackle the conventional share key algorithms problem and avoid the sharing problem by employing two keys.
- Symmetric key algorithms include the Data Encryption Standard (DES), the Advanced Encryption Standard (AES), and Blowfish. Examples of asymmetric key algorithms include RSA and DSA [9]. Many encryption algorithms, such as AES, Blowfish, RC5, RSA, and IDEA, have been modified and are widely used to retain binary data or binary text files [10]. This paper will present a literature review of the Blowfish algorithm and the standard steps. Then this algorithm will be improved and applied to data to improve its efficiency and display the results.

2. Related Work

Compared to other symmetric ciphers, AES and Blowfish provide higher throughput, as mentioned in several papers [11, 12]. As a result, researchers published many papers on the blowfish cipher algorithm. This section will show several of these research papers: In 2011, Afaf proposed a new method to develop the blowfish algorithm. She implemented it by adding a key and replacing the old XOR with a new operation (#) by using two keys in each round of the original Blowfish algorithm. To produce the next left part, the xL (i.e., left part of plaintext) and Pi will be used with the first key K1. The xR (right part of plaintext) and F (xL) will be used with the second key K2 to generate the next right part. She has implanted three keys using one-dimensional cellular automation [13].

In contrast, in 2012, Rohilla evaluated the existing ways applied to the Blowfish algorithm for enhancement to produce a high level of performance [14]. Singh used the Blowfish algorithm to encrypt and decrypt an image using a 64-bit secret key in 2013. The proposed algorithm was implemented using MATLAB to improve security and performance [15]. In the same year, Alabaichi et al. attempted to strengthen the security of the 64-bit Blowfish algorithm by raising both block size and key length in the same year. By extending the key size to 112 bytes rather than 56 bytes, the 128-bit Blowfish algorithm improved the security of the standard algorithm by increasing the complexity of brute force attacks and expanding the block size to 128 bits [16]. Also, Alabaichi et al. attempted to analyze the security of the Blowfish algorithm proposed in [15]. This is done by analyzing the randomness of the output and using the avalanche criteria and correlation coefficient. The results provide an excellent nonlinear relationship between plaintext and ciphertext and a modest avalanche effect from the second iteration [17]. F-function was modified in 2014 by Christina and Joe. Instead of four s-boxes, they used two [18]. Also, Patel and Kamboj developed a strategy to improve the security of the Blowfish block cipher in 2016. The overall number of blowfish rounds is changed in the proposed method by varying the number of blowfish rounds. The outcome of this strategy is to add extra Blowfish cipher security against brute-force attacks by changing the key size used in the Blowfish algorithm.

Furthermore, the suggested approach reduces the blowfish cipher's execution time [19]. In contrast, Ross and Josephraj achieved a parallel processing technique in 2017 by modifying the Feistel F-function of Blowfish using the Runge-Kutta (RK) method. This technique applies to most consumer electronics products that deal with information storage, data transfer, and communication, helping to ensure data security [20].

On the other hand, Quilala et al. modified the Blowfish algorithm. The most significant changes carried out in this modification are a 128-bit block size and a 128-bit key size. Another significant improvement is to reduce the number of S-boxes to two rather than four [21]. In 2019, the update continued on F-functions to improve the Blowfish algorithm. Rekha and Krishnamurthy proposed a method for reducing the number of rounds. Instead of 16 rounds, this strategy was employed for nine rounds. This modification was announced to save time. A change also used dynamic substitution to provide improved security [22]. Finally, G. Quilala and L. Quilala 2021 proposed a new scheme to improve and evaluate the encryption operation in a modified blowfish algorithm. The revision maintained the initial framework, procedure, and use of two S-boxes in the MBA. However, the inserted function offered two derivation procedures to avoid symmetry. Also, they used the avalanche effect and time efficiency to prove the efficiency of the proposed algorithm [23].

3. Theoretical Background

3.1 Text encryption

Text cryptography is an essential tool for providing security for the systems, as well as helping us to identify objects and provide information integrity for the systems [24]. The general structure of the text cryptography framework is shown in Figure 1.

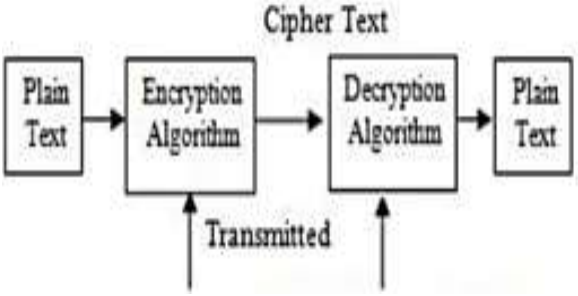

Figure 1: Text cryptography [24]

Two types of cryptography can be used for any encryption system: asymmetric key (public key) cryptography, such as RSA and ElGamal, and symmetric key (secret key) cryptography, such as RC4, AES, DES, and Blowfish [25], as shown in Figure 2. In this paper, the standard Blowfish algorithm is discussed, and several modifications are made to its steps.

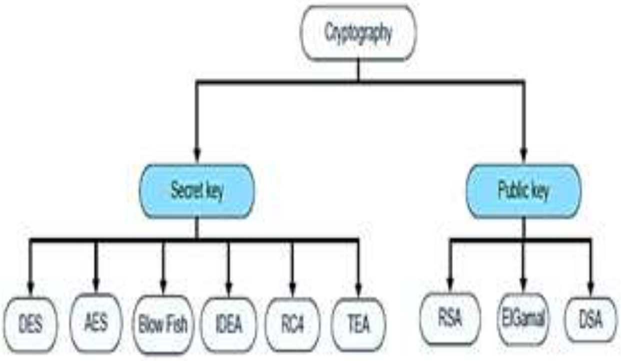

Figure 2: Cryptography types [24]

3.2 Standard Blowfish Algorithm (BA)

Blowfish is a symmetric block cipher created by Bruce Schneier in 1993 as a free and fast alternative to existing encryption techniques. Blowfish is a Feistel block cipher (which divides the input data into fixed-length blocks during encryption and decryption) with a 64-

bit key size, which means data is encrypted and decrypted in 64-bit chunks over many rounds. Before proceeding to the next block, each block is fully encrypted/decrypted. Blowfish operations were done using the shared secret key. It uses a variable-length key ranging from 32 to 448 bits, with 128 bits being the default [7, 19, 26-28]. The structure of this algorithm for the encryption process is shown in Figure 3.

Blowfish uses a large number of subkeys. Therefore, these keys must be precomputed before data encryption or decryption. The essential expansion of BA begins with the P-array and S-boxes and the utilization of many sub-keys, which require precomputation before data encryption or decryption. Below is the explanation of how these sub-keys are calculated [14]:

1 The P-array consists of 18 32-bit subkeys:
P1, P2..., P18.
2 There are four 32-bit S-boxes with 256 entries each:
S1,0, S1, 1..., S1,255;
3 S2,0, S2, 1..., S2,255;
4 S3,0, S3, 1..., S3,255
5 S4,0, S4, 1..., S4,255

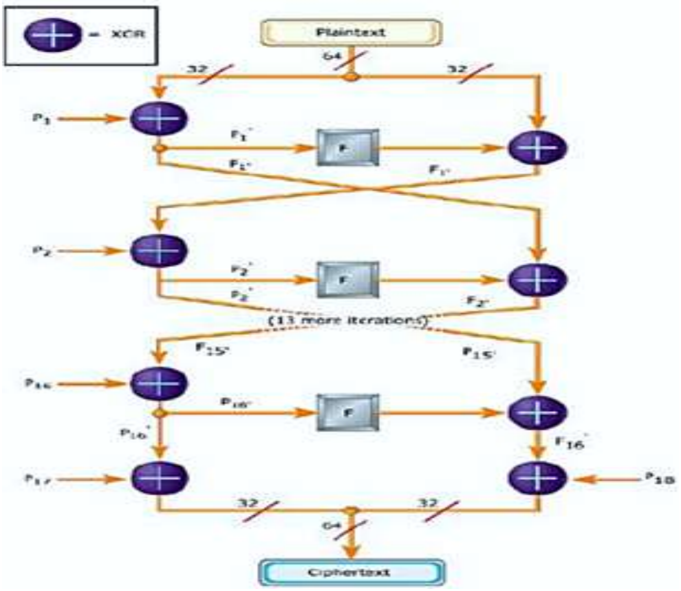

Figure 3: Blowfish algorithm [21]

3.3 F-function

Blowfish is a Feistel network consisting of 16 rounds. The input is a 64-bit data element, x.

```
To encrypt:
 Divide x into two 32-bit halves: xL, xR
For i = 1 to 16:
      xL = xL • Pi
      xR = F(xL) • xR
       Swap xL and xR
Swap xL and xR (Undo the last swap.)
 xR = xR • P17
xL = xL • P18
Recombine xL and xR
```

Function F is as follows as shown in Figure 4:

Divide xL into four eight-bit quarters [20]:
a, b, c, and d F(xL) = ((S1, a + S2, b mod 232) • S3, c) + S4, d mod 232

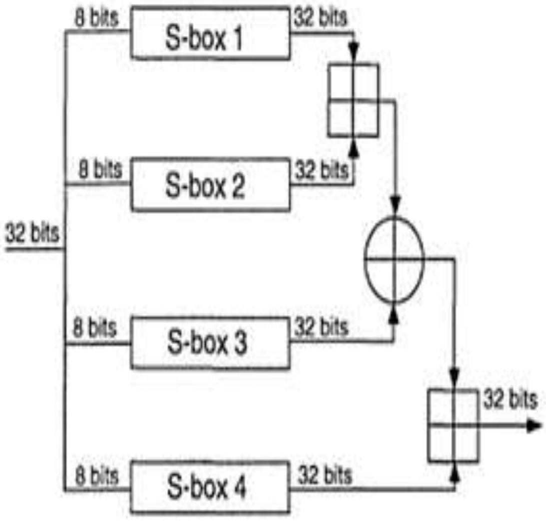

Figure 4: F-Function [19]

3.4 Block Cipher Modes of Operation

Different cryptographic mechanisms could be built by using block ciphers. Encryption algorithms have a property that can be utilized for many other tasks. The list of some of its uses includes:

- Different encryption schemes,
- Stream ciphers,
- Pseudo-random number generator (PRNG),
- Hash functions,
- Message authentication codes (MACs), etc.

Block cipher modes can be defined as combining some simple operations with different ways of using a block cipher for encryption. There are several modes of operation, as illustrated in Figure 5. The deterministic and probabilistic encryption modes represent the main groups of these modes [30].

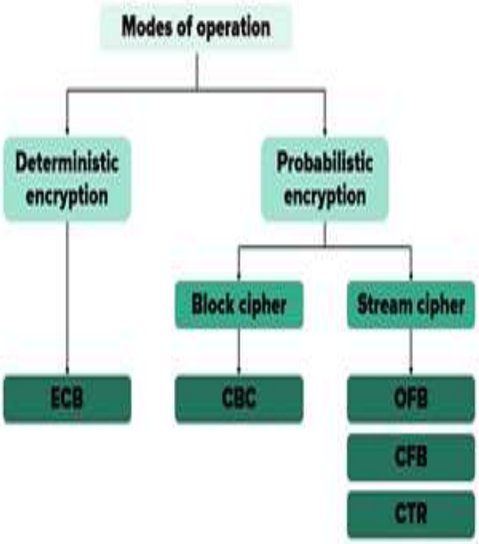

Figure 5: Modes of operation [28].

When the mapping of plaintext to a fixed ciphertext does not change, the scheme is deterministic encryption. While the main characteristic of probabilistic encryption schemes is using randomness to achieve a nondeterministic generation of ciphertext, these schemes can function as both block and stream ciphers. Providing and maintaining confidentiality and authenticity during communication is considered the main goal of all these operation modes. However, security is provided by the cipher rather than the mode [30].

3.5 Output Feedback (OFB) mode

Since block ciphers can be used as stream ciphers, as mentioned previously in this paper, these schemes use the block cipher as a keystream generator, as shown in Figure 6. The initialization vector (IV) is the first input to the block cipher (seed) [31].

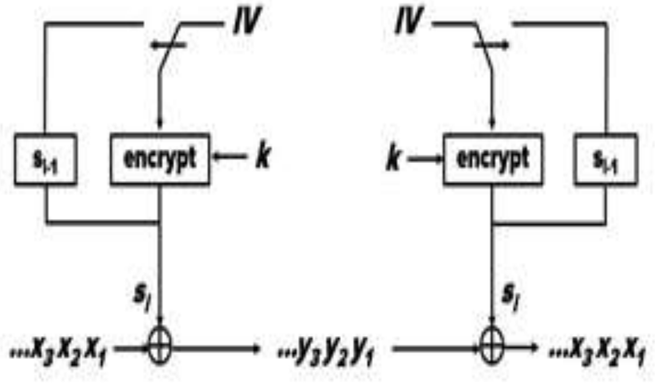

Figure 6: Output Feedback (OFB) mode [30]

The n-bit plaintext is XORed with the n-bit key stream generated during the encryption operation to produce an n-bit ciphertext. The previously generated key streams for the block cipher are fed into the generation of new ones. It is obviously because of this construction that the scheme produces streams block-wise rather than bit-wise.

The block cipher runs as a synchronous stream cipher in the Output Feedback (OFB) mode, which makes it very similar to standard stream ciphers. Keystream generation does not affect either plaintext or cipher text. As presented in Eqs. (1) and (2), operations are exactly the same because using the XOR function during encryption is reversed by another XOR function during decryption. This mode has a major advantage, represented by the feedback mechanism that can work offline before the arrival of the data. On the other side, since each keystream depends on all previous key streams, encryption and decryption cannot be parallelized [31].

encryption: s1=e(IV); y1= (s1 XOR x1) si=e(si-1); yi= (si XOR xi), i≥2 (1)

decryption: s1=e(IV); x1= (s1 XOR y1) si=e(si-1); xi= (si XOR yi), i≥2 (2)

4. Research Methodology

The proposed optimized blowfish algorithm is explained in this section and is depicted in Figure 2, a block diagram showing the transmitter and receiver operations. The expanded key is the first step to modifying Blowfish. First, it expanded to 512 bits (64 bytes). Next, a key schedule is produced by transforming these bits into a 64-byte array. Every 4-rounds, a key schedule is cycled after being accessed as a 128-bit segment. During initialization, a state vector is also generated as 128-bits, like the Blowfish s-boxes.

Before incrementing to the next segment that is used to seed the encryption function, the dynamic update of Ambit's shift, register/key schedule with the output of the encryption function is achieved. It is accomplished by hashing the output of the encryption function with the current key segment. After a side-shift cycle is completed over four rounds, the segment is hashed with the state. Then, the dissociation function transforms the previous result. The ciphertext is produced by hashing the plaintext on the first pass (i.e., the current block of input data) with the state (i.e., the output of the encryption function). The state is also fed back into the key schedule. In every round, the state is hashed with each block of input. The product of the key is simply the state, and the state doesn't affect the input data. Therefore, transmission errors don't continue. The most important differences between the standard blowfish algorithm and the proposed algorithm are summarized in Table 1.

Table 1: Comparison of algorithms Properties

Algorithm	Block Size in bits	Key size in bits	Round Numbers
Standard Blowfish Algorithm	64	32-448	16 based-on S-Box
Optimized Blowfish Algorithm	128	512	16 based-on OFB

In the proposed algorithm, the text input is broken into blocks of equal length (64-bit width for each). Algorithm 1 shows the main steps of the proposed algorithm.

```
Algorithm1: the optimized Blowfish algorithm.
   Step1: Input the Plaintext.
   Step2: Input the key seed data.
   Step3: Convert the Plaintext to equivalent binary data.
Step4: Divide the binary data into blocks; each one has 128-bit.
Step5: Expand the key to 512 bits (64 bytes) accessed as 128 bits. So, the key is cycled every 4 rounds and works in the same behaviour as in S-boxes in a standard blowfish algorithm.

Step6: For each block, do
Generate a sub-key for the encryption function.
Result= XOR (sub-key, input block)
Get the hash value from the current sub-key to produce the cipher and feed it back into the key schedule.
Transform the Result using the Dissociation function.
```

5. Results and Analysis

To ascertain the efficiency of the modified Blowfish algorithm, several tests were conducted. As shown in Table 2, two input messages were used with different sizes of keys. Even if we are using the same input message, when the encryption key is changed, the input message will be affected and completely changed. To prove that the entered text was affected by the key, the Hamming distance was calculated by comparing the differences between the ciphertext generated before and after the key change.

Then the Avalanche effect was measured (see Eq. (1) [31]), as shown in Table 3.

$$Avalanche\ effect = \frac{hamming\ distance}{size} \tag{1}$$

On the other hand, to clearly show the superiority of the proposed algorithm and to evaluate the proposed algorithm's efficiency, this paper considers a dataset with different-sized files (only English data) to compare with Quilala’s proposed algorithm [22]. The implementation results are shown in Tables 4 and 5 for encryption and decryption operations, respectively, which present the efficiency and flexibility of the proposed method.

As shown in these tables, the waiting time of the proposed algorithm is stable and has a significant value. It can also be noted that the encryption time is a little more than the time it takes to decrypt the message to get the original text. However, the proposed algorithm's time varies from low to medium, and it is still faster than the MBA1 and MBA2, with an acceptable Avalanche effect, demonstrating the efficiency of the proposed method.

Table 2: The plaintext messages and the input keys used to test the modified algorithm

Text in bits	Input Text	Key length	Input key	Encryption	Decryption
144	123444 abcfd	208	youhavethemostsecuresystem	_…õP©Jô°´□»•	123444 abcfd
		240	youhavethemostsecuresystemcomp	2íÀ: :#Áyº+ÿþ¼šC	
		272	youhavethemostsecuresystemcomputer	Mƒz(¦mUT8²²v v?.	
		512	theseattackshavemoredifficultsyouhavethemostsecuresystemcomputer	™ó½ˆ×fx ¸W+Jz†¹	
168	safa sami abduljabbar	208	youhavethemostsecuresystem	Ì7ßA.†Ù8+□ªí…□›¡]…·Í]Ë	safa sami abduljabbar
		240	youhavethemostsecuresystemcomp	•©Ò • - Bn0ÛFÚ²I,«/µÃ	
		272	youhavethemostsecuresystemcomputer	– Vª}õ)2%LCŸ}üyê□ !¡-{	
		512	theseattackshavemoredifficultsyouhavethemostsecuresystemcomputer	b„‰IJ™v¢Ñé¼¿ãn Ác°:¾"	

Table 3: The Avalanche effect of two plaintext messages using five keys

Input text in bits	Hamming distance	Avalanche effect
144	1	69%
168	1	59%

Table 4: The Encryption Waiting Time(milliseconds) for the proposed algorithm compared to [22]

Input Size (kB)	Encryption Time (ms)		
	MBA Derivation1	*MBA Derivation2*	*Proposed Algorithm*
10	81.55	49.75	15
20	139.60	84.90	16
50	321.50	196.40	62
100	625.00	379.80	141

200	1210.90	742.20	203
500	2984.00	1831.40	453
1000	5991.85	3432.20	922

Table 5: The Decryption Waiting Time(milliseconds) for the proposed algorithm compared to [22]

Input Size (kB)	**Decryption Time (ms)**		
	Decryption Time (ms)	**Decryption Time (ms)**	**Decryption Time (ms)**
10	142.55	93.15	5
20	267.95	163.35	10
50	626.45	385.95	47
100	1229.55	752.20	125
200	2412.20	1471.70	156
500	6002.50	3670.70	406
1000	12049.15	7274.45	766

5. Conclusions

The first tool that can be used to defend against intrusion in data security is cryptography. This tool is divided into two major categories that serve security goals: symmetric and asymmetric cryptography. The blowfish encryption algorithm is one of the most widely used symmetric algorithms. The suggested approach increases the number of bits in both block sizes (from 64 to 128) and key sizes (from 32-448 to 512) to increase the security level of the Blowfish block cipher. The algorithm was also improved by dropping the S-Box and implementing the OFB strategy in every round. The outcomes demonstrate that the suggested method increases Blowfish security and efficiency, which are assured via multi-pass, while the attackers cannot use superior speed. Also, by deleting any associations that might exist between cipher blocks that are individually encrypted, OFB mode increases security like other modes like cipher block chaining (CBC). Another benefit of OFB is that transmission problems didn't affect the remaining data; therefore, transmission errors don't perpetuate. A CRC checksum is also included in the proposed algorithm for validating the decryption operation.

6. Acknowledgements

We would like to thank the staff of the University of Baghdad for providing the materials that enabled us to complete this research.

7. Conflict of Interest

The authors declare that they have no conflicts of interest.

References

[1] R.R. Asaad, S.M. Abdulrahman, and A.A. Hani, "Advanced Encryption Standard Enhancement with Output Feedback Block Mode Operation", *Academic Journal of Nawroz University*, vol. 6, no. 3, pp.1-10, 2017.

[2] A. G. ,Naser, F. A.H. ,Majeed, “Constructing of AnalysisMathematicalModel for Stream Cipher Cryptosystems”, *Iraqi Journal of Science*, vol. 58, no.2A, pp. 707-715, 2017

[3] F.F., Saleh and N.H.M., Ali. “Generating Streams of Random Key Based on Image Chaos and Genetic Algorithm”. *Iraqi Journal of Science*, vol. 36, no. 8, pp. 3652-3661, 2022.

[4] W. Stallings, *Cryptography and network security principles and practices, 4th edition, ISSN: 0131873164, Publisher:Prentice Hall*, 2006.

[5] R.B. Marqas, S.M. Almufti, and R.R. Ihsan, "Comparing Symmetric and Asymmetric cryptography in message encryption and decryption by using AES and RSA algorithms", *Xi 'an Jianzhu Keji Daxue Xuebao/Journal of Xi'an University of Architecture & Technology*, vol. 12, pp.3110-3116, 2020.

[6] G.J. Simmons, "Symmetric and asymmetric encryption", *ACM Computing Surveys* (CSUR), vol. 11, no. 4, pp.305-330, 1979.

[7] A. Jain, and V. Kapoor, "Secure communication using RSA algorithm for network environment", *International Journal of Computer Applications*, vol. 118, no. 7, pp. 6-9, 2015.

[8] S. Chandra, S. Paira, S.S. Alam, and G. Sanyal, " A comparative survey of symmetric and asymmetric key cryptography", In *international conference on electronics, communication and computational engineering* (ICECCE), pp. 83-93. IEEE, 2014.

[9] A. Singh and S. Malik, "Securing Data by Using, Cryptography With Steganography", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 5, 2013.

[10] A.T. Hashim, A.H. Jassem, and S.A. Ali, "A Novel Design of Blowfish Algorithm for Image Security", In *Journal of Physics: Conference Series, IOP Publishing*, vol. 1818, no. 1, p. 012085, 2021.

[11] P. Patil, P. Narayankar, D.G. Narayan, and S. M. Meena, "A comprehensive evaluation of Cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish", In: 1st *International Conference on Information Security & Privacy, Procedia Computer Science*, vol. 78, pp. 617–624, 2016.

[12] A. Ramesh, and A. Suruliandi, "Performance analysis of encryption algorithms for information security", *IEEE International Conference on Circuits, Power and Computing Technologies* (ICCPCT), pp. 840–844, 2013.

[13] A. Al-Neaimi, and R. F. Hassan, "New Approach for Modifying Blowfish Algorithm by Using Multiple Keys", *International Journal of Computer Science and Network Security*, vol. 11, no. 3, 2011.

[14] P. Rohilla, "Blowfish Algorithm: Security and Performance Enhancement", *World Academy of Informatics and Management Sciences*, vol. 1, no. 5, 2012.

[15] P. Singh, and K. Singh, "Image encryption and decryption using blowfish algorithm in MATLAB", *International Journal of Scientific & Engineering Research*, vol. 4, no. 7, pp.150-154, 2013.

[16] A. Alabaichi, R. Mahmood, F. Ahmad, and M. Mechee, "Randomness Analysis on Blowfish Block Cipher Using ECB and CBC Modes", *Journal of Applied Sciences*, vol. 13, pp. 768-789, 2013.

[17] A. Alabaichi, F. Ahmad, and R. Mahmod, "Security analysis of blowfish algorithm". In *Second International Conference on Informatics & Applications* (ICIA), IEEE, pp. 12-18, 2013.

[18] L. Christina , J. Irudayaraj, "Optimized Blowfish Encryption Technique", *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 7, 2014.

[19] R. Patel, and P. Kamboj, "Security enhancement of blowfish block cipher", In *International conference on smart trends for information technology and computer communications*, Springer, pp. 231-238, 2016.

[20] B. S. Ross, and V. Josephraj, "Performance Enhancement of Blowfish Encryption Using RK-Blowfish Technique", *International Journal of Applied Engineering Research*, vol. 12, pp. 9236-9244, 2017.

[21] C. Rekha, G. N. Krishnamurthy, "An Optimized Encryption Algorithm and Function With Dynamic Substitution For Creating S-Box And P-Box Entries For Blowfish Algorithm", *International Journal of Science & Technology Research*, vol. 8, no. 12, 2019.

[22] T. F. Quilala., and R. L. Quilala, "Modified Blowfish algorithm analysis using derivation cases", *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2192-2200, 2021.

[23] A. Ghorpade, and H. Talwar, "The Blowfish Algorithm Simplified", *International Journal of Advanced Research in Electrical,Electronics and Instrumentation Engineering*, vol. 5, no. 4, 2016.

[24] I. Latif, "Time evaluation of different cryptography algorithms using labview", In *IOP Conference Series: Materials Science and Engineering*, vol. 745, no. 1, p. 012039. IOP Publishing, 2020.

[25] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (Blowfish)", In *International Workshop on Fast Software Encryption*, Springer, pp. 191-204, 1993.

[26] T. Nie, and T. Zhang, "A study of DES and Blowfish encryption algorithm", In *Tencon 2009-2009 IEEE Region 10 Conference*, *IEEE* , pp. 1-4, 2009.

[27] M.N. Khatri–Valmik, and V.K. Kshirsagar, "Blowfish algorithm", *IOSR Journal of Computer Engineering* (IOSR-JCE), vol. 16, no. 2, pp.80-83, 2014.

[28] R. Anderson, E. Biham, and L. Knudsen, “Serpent: A Proposal for the Advanced Encryption Standard”, *First Advanced Encryption Standard (AES) Conference*, Ventura, CA, 1998.

[29] National Institute of Standards and Technology. DES Modes of Operation, FIPS PUB 81(1998).

[30] D. Bujari, and E. Aribas, “Comparative Analysis of Block Cipher Modes of Operation”, *International Advanced Researches & Engineering Congress, Osmaniye, Turkey*, pp. 16-18, 2017.

[31] R. Ariel, E. Festijo, and R. Medina, "Blowfish-128: a modified blowfish algorithm that supports 128-bit block size", In *8th International Workshop on Computer Science and Engineering, Bangkok, Thailand*, pp. 578-584, 2018.