American Journal of
Biometrics & Biostatistics

Original Article

An Enhanced Blowfish Algorithm with Reduced Computational Speed

Gbolagade Morufat Damola[1]*, Rasheed Jimoh[2] and Oluwakemi Abikoye[2]

[1]*Department of Computer Science, Al-Hikmah University, Ilorin, Nigeria*
[2]*Department of Computer science, University of Ilorin, Nigeria*

***Address for Correspondence:** Gbolagade Morufat Damola, Department of Computer Science, Al-Hikmah University, Ilorin, Nigeria, Tel: +234-803-234-3270; E-mail: dammyconsult@gmail.com

Submitted: 16 February 2022; **Approved:** 12 May 2022; **Published:** 13 May 2021

Cite this article: Damola GM, Jimoh R, Abikoye O. An Enhanced Blowfish Algorithm with Reduced Computational Speed. American J Biom Biostat. 2022 May 13;5(1): 001-07.

ABSTRACT

Cryptography guarantees security in communication through encryption. By superimposing shares in pairs, it is possible to identify any instances of fraud. The hidden picture is retrieved, and its contrast has been multiplied by (\frac {m}{m+1}\) to return to the original contrast. Contrast restored is (\frac {1}/{2(m + 1)}). Unlike normal (k,n)-VC approaches, these schemes do not address their flaws. They need more shares, bigger pixels, or lower contrast. This is because these schemes do not address their flaws. The experimental results show that the concealed quality of the picture corresponds to expectations [1] in a paper that points out that presently, data exchange is a need for people to communicate. When sending data on the Internet, various security risks must be addressed. There are three parts to the research. The first part, an algorithm of 9 × 6 matrices for fair play, is a new proposal for gameplay called the RSA algorithm. A combination of algorithms strengthens symmetric and asymmetric algorithms despite producing good results in various parameters, such as message strength and key strength. This paper proposed modified Blowfish encryption that uses 128-bit block size,128-bit key, and a new F- function formula was derived.The modification was also done on the original structure using the # function to enhance the security strength to replace XOR. Time and avalanche were used to assess the algorithm's performance. Due to difference in block sizes, the modified Blowfish is slower with average key, encryption, and decryption times of 26.99 ms, 1651.83 ms, and 2765.04 ms, respectively, compared to Blowfish with 21.65 ms, 1297.76 ms, and 2176.59 ms. Applying a 128-bit block size enhances security by reducing the chances of having duplicate blocks that may reveal the information. With encryption and decryption average times of 2418.08 ms and 4002.70 ms, respectively, the improved Blowfish is quicker than Twofish. The different derivations enhanced the modified Blowfish's avalanche. Blowfish had a 95.14 percent success rate, whereas modified Blowfish had a 99 percent success rate.

Keywords: Algorithm; Avalanche; Blowfish; Modify blowfish; Security

INTRODUCTION

Since the Internet and communication networks are rapidly growing, the security and privacy of users are challenged with great feelings of vulnerability. In contrast, Internet cryptology is the defence technique or anonymity of writing or reading and delivering messages in the encrypted form [1,2]. Many cryptographic algorithms have been proposed and applied to provide consumers with safe internet-based transactions. However, with society's hacking rate increasing, it seems additional study efforts are needed. The latest rampant hacking encounters have made existing cryptographic algorithms no longer secure [3-7]. The measurement of encryption assumes a vital role in protecting knowledge as it is substituted or shared. There are two styles of encryption algorithms: symmetric key encryption algorithms and key encryption algorithms. While asymmetric, Symmetric Key Encryption or secret key encryption algorithm requires two keys, uses the same key to encrypt and decrypt.

Literature review

[8,9] postulated that Blowfish is a vector. The algorithm is divided into two parts: critical expansion and data encryption. A 448-bit key is extended into 4168-byte arrays by crucial development. The data were encrypted using Feistel's 16-round network. Each round includes key-dependent permutation as well as data-dependent substitution. Both 32-bit XORs and attaches. The only extras are four round-indexed data lookups. Using a 64-bit block size (compared to, say, AES' 128-bit block size) makes it vulnerable to attacks, especially in contexts like HTTPS [10]. Because of its limited block size, it was not advised to encrypt 4GB of data.

According to John Kelsey, an attack developed by the researcher could break 3-round Blowfish, whereas he could not expand it. When Fis is known, differential cryptanalysis can reveal all other keys with 248 plain texts selected against the number of rounds when restricted to round eight. However, it is exceedingly difficult for Florence's more significant number. Despite that, blowfish are extraordinarily known as the best block cipher in terms of speed; increasing the number of bits will enhance the performance [11-14]. Two fish encryption algorithm based on Blowfish takes128 bits block size as input was admitted and seen as the best algorithm compared with AES [15,16]. It generates high-security strength but has low speed as compared with Blowfish. Two fish is regarded as a well-known usage compared with Blowfish [17]. Blowfish has two parts critical expansion and data encryption. The critical expansion parts applied XOR to the variable key length. The plain text is also used to produce subkeys in which four critical independent s-boxes are generated. Every round needs four KB, making the algorithm unsuitable for several devices with small memory, such as smart cards and phones. Using the said techniques, computing subkeys in each round fallout to a slower operation, making it efficient when using an application that needs changing secret key regularly [18,19]. At the same time, three possible simplifications recommended by [20-22] aimed at increasing the security strength without affecting the execution speed. Most researchers concentrate more on reducing execution time by improving Blowfish's key generation to produce subkeys [22]. However, the outcomes of the researchers were able to achieve a slight time reduction in time complexity which may harm the security strength. Some researchers have proposed the extension of the block size of Blowfish to 128- bit [23]. However, the findings suggest that additional study is needed to boost speed while complying with the security strength requirement. This research presents a modified blowfish approach with the fewest possible iterations.

The following objectives were explored: to compare the performance of the modified Blowfish, Blowfish, and Twofish techniques in terms of encryption. From there, a modified blowfish method is presented, which uses a 128-bit block size and a modified fiestal iterative structure to improve performance and a new F-function to improve security. Decryption speed; and compare the performance of the modified algorithm and Blowfish in terms of security utilizing the avalanche effect. This research will modify Blowfish to employ a 128-bit block size and a 128-bit key as established in Figure 1. The increase in block size would allow for file encryption with a lower risk of duplicate blocks. The original Blowfish structure will be retained, but the number of boxes will be reduced from four to two to reduce memory consumption.

The algorithm shows:

1. Blowfish's 16 rounds.
2. The input is a 64-bit element, x.
3. Divide x into two 32-bit halves.

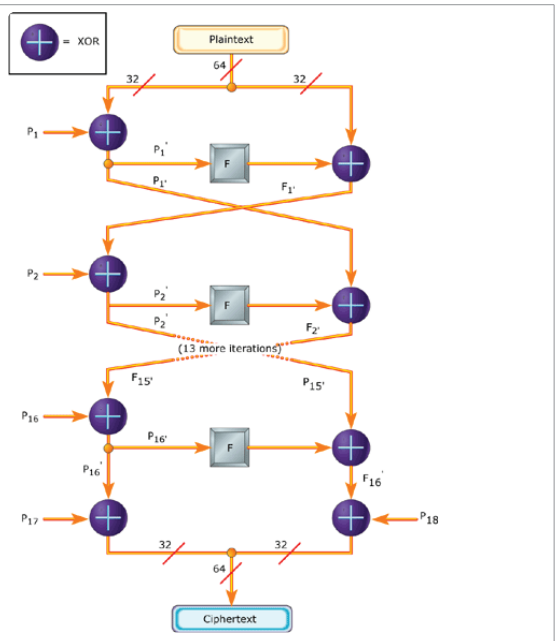

Figure 1: Existing blowfish Iteration process algorithm source [23,24].

4. Then, for I = 1-16:

xL = xL Pi XOR

F(xL) XOR XR

5. Swap XL/XR

6. After round sixteenth, exchange xL and xR to reverse the previous swap.

xR = xR XOR P17 and xL = xL XOR P18.

7. Recombine XL and XR to get the ciphertext.

8. Decryption is the same as encryption, except that P1, P2,..., P18 are used in reverse sequence.

9. Blowfish implementations requiring the highest speeds can unroll the loop and ensure all subkeys are placed in a cache.

Summary of related works

[10,11] stated that various applications, such as multimedia transmission and data storage use VC-based security systems. It takes one encrypted picture and makes it seem to be several unsecured pictures via a process known as "printing transparencies." It is pretty simple to decode the secret because it does not necessitate any cryptography understanding of computation. However, the researchers found that fraudulent shareholders are highly likely to present falsified shares during the covert reconstruction phase, causing significant harm to honest shareholders. The researchers [12] [24-26] proposed a secure method for verifying cheating shares to accomplish a fair picture secret reconstruction. It was created to allow the original shareholders of the XOR-based VC method to exchange a verification picture. The pixel expansion is raised by one to accomplish the verification function.

The algorithm depended on the algorithm, and a more complex structure resulted in poor performance time. It was recommended that more research be done on the blowfish algorithm.

METHODOLOGY

The suggested approach is a resilient secret-key block cipher that boosts reliability by boosting intakes and altering the F-function of the present Blowfish. This research enhances efficiency without compromising the established Blowfish algorithm's memory, security, and simplicity.

Enhanced blowfish iteration process

A new way to manipulate bits has been shown that uses a different truth table to manipulate bits that work on 4-states to make intruders' encryption methods more secure and keyspace bigger (0,1,2,3). Only the bits (0,1) are used in (XOR). The # symbol was used to refer to the operator in the truth tables seen in tables 1-4.

The new operation requires three inputs. A cross-point is defined as the intersection of two rows and columns in a table.

The proposed update utilizes the current procedure hash function (#) introduced in the original Blowfish algorithm during each round. An additional key is needed to apply this operation on both sides, a binary key that transforms into a 4-state key. The first K1 key will be used with XL and Pi to produce the next.

She left part in each round of the initial Blowfish. The second key will use F(XL) and XR to create the correct position. All three inputs can be transformed from 64 bits to 32 digits, each of which can be one of four states (0, 1, 2, 3), i.e., converted to the corresponding decimal digits.

Table: 1

#0	0	1	2	3
0	3	2	1	0
1	2	3	0	1
2	3	0	1	2
3	0	1	2	3

Table: 2

#1	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Table: 3

#1	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Table: 4

#3	0	1	2	3
0	1	0	3	2
1	0	1	2	3
2	3	2	1	0
3	2	3	0	1

The inputs should be ordered in this sequence to match the result in the table above.

Input 1: Input key/ password supplied by the user

Input 2: P-Array, gotten from the hexadecimal binary of pi

Input 3: half of the input 128 bits, i.e., XL

The new operation requires three inputs. The first input specifies the table number; the other two inputs provide the row and column numbers in the chosen table where the cross-point provides the result. An example of service is seen below:

Input 1: 0 1 3 1 2 3 1

Input 2: 3 2 1 0 1 1

Input 3: 1 0 0 2 1 2

——————————————————-

Result: 1 0 0 1 2 3 2

It takes the ORDER KI, PI, XL

Where K1=b2 bit key, Pi=p array,xl=first 32bit input from the left

XL = 00101010010101011101010010111110111 000010101001010 10111010100101111101110 0

Moreover, the value of Pi-1, which represent here the first key, could be the binary number:

Pi = 10010100011101010101001001110100110 100101000111010 10101001001110100110 1

and the entered key, which represent the second key in the applied # operation, the binary number:

K1=11101010010101011010111010101010010111 11101010010101 011010111010101010010111

Firstly, the three entered 32-bits binary numbers should be converted to a 4-states 16-digit numbers.

Pi ‘ = 2 1 1 0 1 3 1 1 1 1 0 2 1 3 1 0 3 1 2 1 1 0 1 3 1 1 1 1 0 2 1 3 1 0 3 1

xL ‘ = 0 2 2 2 1 1 1 1 3 1 1 0 2 3 3 1 3 0 0 2 2 2 1 1 1 1 3 1 1 0 2 3 3 1 3 0

K1’ = 3 2 2 2 1 1 1 1 2 2 3 2 2 2 2 1 1 3 3 2 2 2 1 1 1 1 2 2 3 2 2 2 2 1 1 3

Then the # operation applied according to table 3,4, 5, and 6, the result of Encryption will be:

New XL = 3 1 1 0 0 2 0 0 0 2 0 0 1 2 0 1 0 0 3 1 1 0 0 2 0 0 0 2 0 0 1 2 0 1 0 0

If we reverse the whole operation, we will get the initial, which is the result of the decryption operation that equal to the original data:

K1’ = 3 2 2 2 1 1 1 1 2 2 3 2 2 2 2 1 1 3 3 2 2 2 1 1 1 1 2 2 3 2 2 2 2 1 1 3

Pi’ = 2 1 1 0 1 3 1 1 1 1 0 2 1 3 1 0 3 1 2 1 1 0 1 3 1 1 1 1 0 2 1 3 1 0 3 1

New XL = 3 1 1 0 0 2 0 0 0 2 0 0 1 2 0 1 0 0 3 1 1 0 0 2 0 0 0 2 0 0 1 2 0 1 0 0

XL = 0 2 2 2 1 1 1 1 3 1 1 0 2 3 3 1 3 0 0 2 2 2 1 1 1 1 3 1 1 0 2 3 3 1 3 0

Existing F- function

Existing function F is as follows:-

Divide X_L into four eight-bit quarters: a, b, c, and d

$S1_{a=}$ first S BOX WITH 8 BITS,$S2_b$ = second SBOX WITH 8 BITS,$S3_C$ = THIRD S BOX WITH 8 BITS AND $S4_d$ = fourth S BOX WITH 8 BITS

$$F\left(X_L\right)=\left((S1_{,a}+S2_{,b}\ mod\ 2^{32}\right) XOR\ S3_{,c})+S4_{,d}\ mod\ 2^{32} \quad (1)$$

Enhance function F2

Without violating the security requirements, the Blowfish function F was modified as follows and shown in Figure 3:

$$F_1\left(xL\right)=\left(\left(\left(S_1 xor\ S_2\right)+S_3, mod\ 2^{64}\right) xor\ S_4\right) \quad (2)$$

$$F_2\left(xL\right)=\left(\left(\left(S_5 xor\ S_6\right) xor S_7\right) xor\ S_8\right) \quad (3)$$

$$F\left(xL\right)=\left[F_1\left(xL\right), F_2\left(xL\right)\right] \quad (4)$$

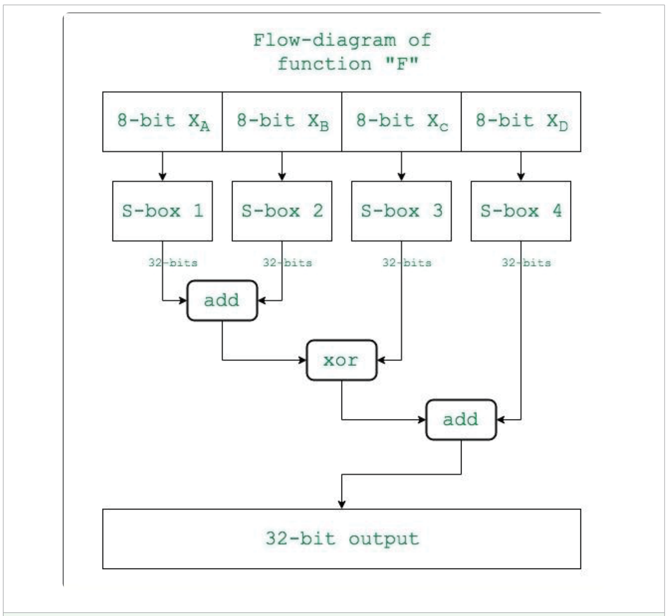

Figure 2: Existing blowfish function F [23-26].

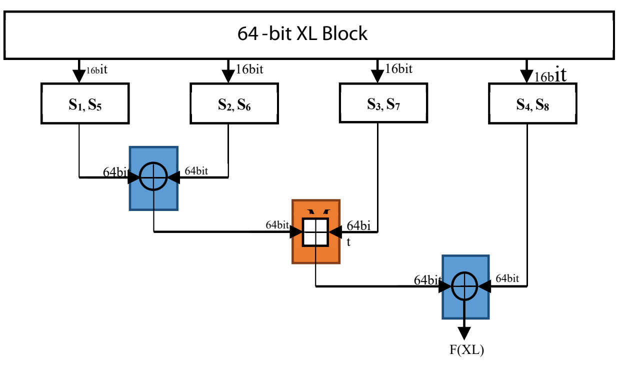

Figure 3: Modified blowfish function F.

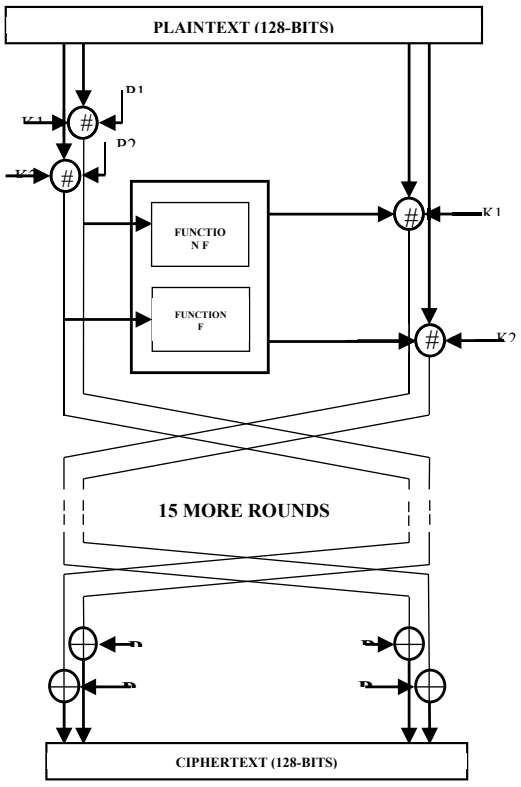

Figure 4: Enhanced blowfish iteration process.

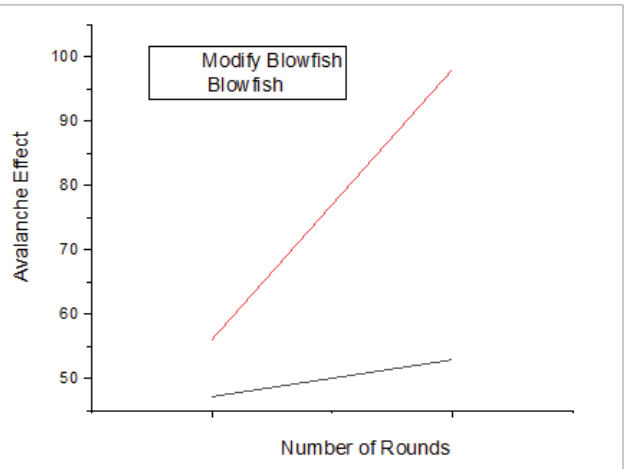

Figure 5: Blowfish impact and proposed modified blowfish.

Generating F functions F1 and F2, each with four S –boxes, concatenating the F1 and F2 gives the final F(xL).

This modification upgrades the original 64 bits Function F to 128 bits Function F, reducing time.

Fiestal structure

In order to improve the security of the blowfish method, two function keys, K1 and K2, are utilized, which the users may use to substitute XOR and provide to the # function. The input size is also increased to 128 bits, making it possible to protect data in a short amount of time.

Significance of enhancing blowfish

1. The processing time of an algorithm is decreased when compared with an original algorithm.
2. Here 1 - ADDITION and 4 XOR are used, but the original F1- function uses 2- addition and 1XOR
3. It takes 128 bits as the input intake, which reduces the number of iterations when large files are encrypted.
4. The #- function to replace XOR in the fiestal process makes it difficult to attack.
5. It improves the security standard.
6. It is complicated for the attackers to realize that the F-functions is modified, and chances of the attacker are slim when compared to the original Blowfish.
7. 8 S- boxes were generated with Two F functions which are hard to cryptanalysis.
8. The avalanche effect states that a minor change in the plaintext (or key) should significantly change the ciphertext.

Avalanche Effect = (Number of Changed bit in ciphertext) / (Number of bits in ciphertext).

A good cipher should always satisfy an avalanche > 50%.

SIMULATED ANALYSIS OF BLOWFISH ALGORITHM

(Table 1) reveals that Modified Blowfish Algorithm (MBA) has an average of 22.10 ms while Blowfish Algorithm (BA) is 27.12 ms. Rounds were reduced to 8 to offset the time gap. While the initial algorithm appears quicker than the updated version, 128-bit block size is still used. Extending block size to 128-bit decreases the risk for dual block leakage, enhancing security. The 64-bit mark is around 32 gigabytes (232 blocks of 8 bytes). A 1TB drive is encrypted with 32 redundant blocks. (Table 5) shows the Comparison speed central generation for various file sizes in milliseconds.

(Table 6) shows the encryption in milliseconds utilizes various file sizes compared with the Two Fish Algorithm (TA).

(Table 7) shows the Decryption period in milliseconds utilizing various file sizes.

An appropriate characteristic of any encryption algorithm is that a minor fundamental change will trigger a wide text discrepancy. Compared to Blowfish, the avalanche effect of modified Blowfish ensured that the algorithm's diffusion was not affected by changes in the F- functions. (Figure 4) indicates the avalanche percentage. It can be deduced from the figure below that our developed Algorithm for Blowfish outperforms the existing algorithm with 99% accuracy. As the number of rounds increases, then the avalanche effect also increases. The higher the number of avalanches, the higher the security; the revised algorithm suggested a more significant avalanche and improved safety. (Figure 5) describes the Blowfish impact and proposed modified Blowfish

(Figure 6) describes the input size with the computation speed for different file sizes. The result is improved for modifying Blowfish than with normal Blowfish.

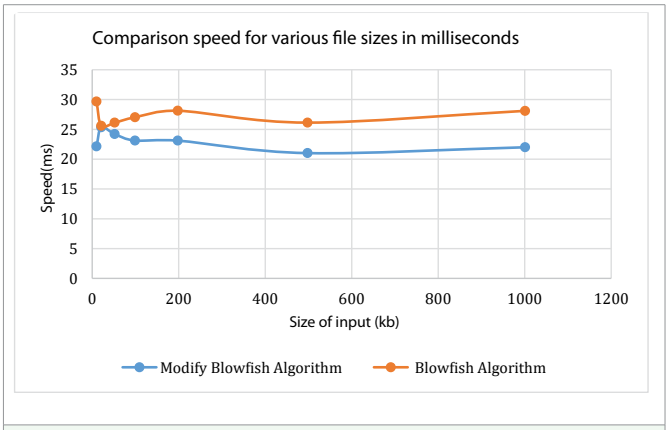

Figure 6: Comparison of speed for various file size.

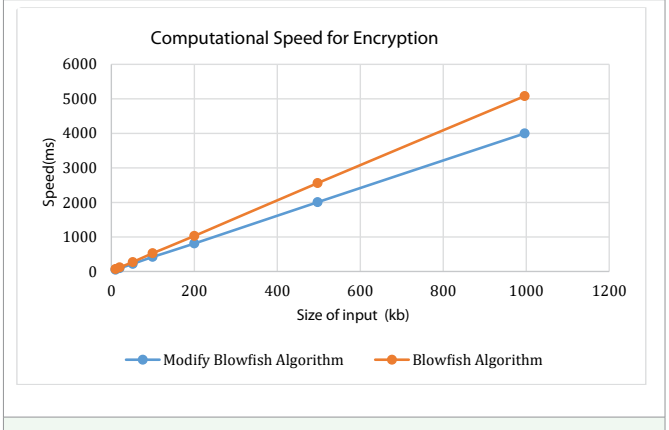

Figure 7: Computation speed for encryption.

CONCLUSION AND RECOMMENDATION

This paper suggests an improved 128-bit block size and 128-bit key Blowfish algorithm, which modifies the initial blowfish architecture with more s-boxes to reduce menarches. Results demonstrate that the modified algorithm remains sufficient for the P-array and S-boxes to achieve original storage performance. Since the modified Blowfish algorithm is faster than the Two algorithms, the reason is the rise in block size and simplicity of the Blowfish Algorithm. The revised Blowfish is quicker and has better output efficiency than Blowfish's related algorithm—two fish. The crypto-based modified Blowfish algorithm has passed entropy and frequency checking. A 128-bit key will take a 5e+025-year brute force attack, making it impossible to use brute force and the algorithm for data security or file encryption experiments. Other researchers may study the modified algorithm for potential hardware optimization work.

REFERENCES

1. Anshul S, Kashif M, Rohit Reddy PB, Ashwin U, Arshad K. Erratum regarding missing Declaration of Competing Interest statements in previously published articles. J Clin Orthop Trauma. 2020 Nov-Dec;11(6):1177. doi: 10.1016/j.jcot.2020.10.025. Epub 2020 Oct 15. Erratum for: J Clin Orthop Trauma. 2019 Mar-Apr;10(2):236-240. Erratum for: J Clin Orthop Trauma. 2019 Mar-Apr;10(2):422-426. Erratum for: J Clin Orthop Trauma. 2019 Jul-Aug;10(4):733-737. Erratum for: J Clin Orthop Trauma. 2019 Jul-Aug;10(4):785-788. Erratum for: J Clin Orthop Trauma. 2019 Jul-Aug;10(4):811-815. Erratum for: J Clin Orthop Trauma. 2019 Sep-Oct;10(5):959-964. Erratum for: J Clin Orthop Trauma. 2019 Sep-Oct;10(5):995-998. Erratum for: J Clin Orthop Trauma. 2019 Oct;10(Suppl 1):S88-S94. Erratum for: J Clin Orthop Trauma. 2019 Oct;10(Suppl 1):S193-S196. Erratum for: J Clin Orthop Trauma. 2020 Mar-Apr;11(2):310-313. Erratum for: J Clin Orthop Trauma. 2020 Mar;11(Suppl 2):S219-S222. Erratum for: J Clin Orthop Trauma. 2020 May-Jun;11(3):504-505. Erratum for: J Clin Orthop Trauma. 2020 May-Jun;11(3):492-497. Erratum for: J Clin Orthop Trauma. 2020 Jul;11(Suppl 4):S428-S430. Erratum for: J Clin Orthop Trauma. 2020 Jul;11(Suppl 4):S448-S455. PMID: 33078051; PMCID: PMC7557265.
2. Law Kumar S, Gupta P. Personal authentication based on iris recognition. Int J Sci Res. 2016;5(2). doi: 10.21275/v5i2.nov161104.
3. Kumar MA, Karthikeyan S. Investigating the efficiency of blowfish and rejindael (AES) algorithms. Int J Comput Netw Inf Secur. 2018;4(2). doi: 10.5815/ijcnis.2018.02.0 4.
4. Bhargavan K, Leurent G. On the practical (in-) security of 64-bit block ciphers: Collision attacks on HTTP over TLS and open VPN. In Proceedings of the ACM Conference on Computer and Communications Security. 2016;24(28). doi: 10.1145/2976749.2978423.
5. Bhargavan K, Leurent G. On the Practical (In-) Security of 64-bit block ciphers. 2016. doi: 10.1145/2976749.2978423.
6. Florence S, Bhuvaneswari Amma NG, Annapoorani G, Malathi K. Predicting the Risk of heart attacks using neural network and decision tree. Int J Innov Res Comput Commun Eng. 2019;2(11). https://bit.ly/3JAb9Bp
7. Gad R, Abd El-Latif AA, Elseuofi S, Ibrahim HM, Elmezain M, Said W. IoT security based on iris verification using multi-algorithm feature level fusion scheme. 2019. doi: 10.1109/CAIS.2019.8769483.
8. Vaudenay S. On the weak keys of blowfish. In lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). 2017;1039(1). doi: 10.1007/3-540-60865-6_39.
9. Kumar V, Kumar R, Barbhuiya MA, Saikia M. Multiple Encryption using ECC and its time complexity analysis. Int J Comput Eng Res Trends. 2016;3(11). doi: 10.22362/ijcert/2016/v3/i11/48907.
10. Naito Y, Sugawara T. Light weight authenticated encryption mode of operation for tweakable block ciphers. IACR Trans Cryptogr Hardw Embed Syst. 2019. doi: 10.46586/tches.v2020.i1.66-94.
11. Pathak M, Srinivasu N, Bairagi V. Effective segmentation of sclera, iris and pupil in noisy eye images Telkomnika. Telecommunication Comput Electron Control. 2019;17(5). doi: 10.12928/TELKOMNIKA.v17i5.12551.
12. Pirbhulal S, Zhang H, Mukhopadhyay SC, Li C, Wang Y, Li G, Wu W, Zhang YT. An efficient biometric-based algorithm using heart rate variability for securing body sensor networks. Sensors (Basel). 2015 Jun 26;15(7):15067-89. doi: 10.3390/s150715067. Erratum in: Sensors (Basel). 2017 Mar 16;17(3): PMID: 26131666; PMCID: PMC4541821.
13. Sasi SB, Sivanandam N, Emeritus. A survey on cryptography using optimization algorithms in WSNs. Indian J Sci Technol. 2021;8(3). doi: 10.17485/ijst/2021/v8i3/59585.
14. Shamim HM, Muhammad G, Rahman SMM, Abdul W, Alelaiwi A, Alamri A. Toward end-to-end biometrics-based security for IoT infrastructure. IEEE Wirel Commun. 2016;23(5). doi: 10.1109/MWC.2016.7721741.
15. Vaudenay S. On the weak keys of blowfish. 2017;27-32.
16. Mahdi JA. Design and implementation of proposed BR encryption algorithm. IJCCCSE. 2018;9(1):1-17. https://bit.ly/3rnXpDs
17. Schneier B, Kelsey J, Whiting D, Wagner D, Hall C. Twofish: A 128-Bit Block Cipher. NIST AES Propos. 1998;15(1):1-27. https://bit.ly/3rJjfBR
18. Muthukumar G, Dharma EG. A comparative analysis on symmetric key encryption algorithms. Int J Adv Res Comput Eng Technol. 2019;3(2):379-383.
19. Atia TS. Development of a new algorithm for key and S-box generation in blowfish algorithm. J Eng Sci Technol. 2019;9(4):432-442. https://bit.ly/3O3cg05
20. Chandrasekaran J, Raman JS. Ensemble of blowfish with chaos based Sbox design for text and image encryption. Int J Netw Secure Its Appl. 2011;3(4):165-173. doi: 10.5121/ijnsa.2011.3415.
21. Abd El-Sadek AA, El-Garf TA, Fouad MM. Speech encryption applying a modified blowfish algorithm. International Conference on Engineering and

Technology. 2019;1-6. doi: 10.1109/ICEngTechnol.2014.7016764.

22. Alabaichi AM. A dynamic 3D S-Box based on cylindrical coordinate system for blowfish algorithm. Indian Indonesian J Elec Eng & Comp Sci. 2015;8(30):1-17. doi: 10.17485/ijst/2015/v8i30/86800.
23. Oishi NJ, Mahamud A, Asaduzzaman. Short paper: Enhancing Wi-Fi security using a hybrid algorithm of blowfish and RC6. International Conference on Networking Systems and Security. 2016;1-5. doi: 10.1109/NSysS.2016.7400706.
24. Alabaichi AM, Mahmood R, Ahmad F, Michel MS. Randomness analysis on blowfish block cipher using ECB and CBC Modes. J Appl Sci. June 2013;13(6):768-789. doi: 10.3923/jas.2013.768.789.
25. Nada Hussein MA, Suaad AA. Modified blowfish algorithm for image encryption using multi keys based on five S-boxes. Iraqi Journal of Science. 2016.
26. Josephraj V. Performance enhancement of blowfish encryption using rk-blowfish technique. International Journal of Applied Engineering Research. 2017;12:9236-9244.