Indonesian Journal of Electrical Engineering and Computer Science
Vol. 12, No. 1, October 2018, pp. 38~45
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v12.i1.pp38-45

Modified Blowfish Algorithm

Theda Flare G. Quilala[1], Ariel M. Sison[2], Ruji P. Medina[3]
[1,3]Technological Institute of the Philippines, 938 Aurora Blvd., Cubao, Quezon City, Philippines
[2]Emilio Aguinaldo College, 1113-1117 San Marcelino St., Paco, Manila 1000, Philippines

Article Info

Article history:

Received Feb 26, 2018
Revised Apr 21, 2018
Accepted Jun 13, 2018

Keywords:

Twofish
Cryptography
Encryption
Security
Cipher

ABSTRACT

Cryptography guarantees security in communication through encryption. This paper proposed a modified Blowfish encryption that uses 128-bit block size and 128-bit key to comply with minimum requirements as an encryption standard. The modification retained the original structure for easy migration but utilized two S-boxes to save memory. A derivation was added to prevent symmetry. The algorithm's performance was evaluated using time, and avalanche. Upon testing, the modified blowfish is slower with key, encryption, and decryption average of 26.99ms, 1651.83ms, and 2765.04ms compared to blowfish with 21.65ms, 1297.76ms and 2176.59ms due to block size difference. Applying 128-bit block size increases security by decreasing the chances of having duplicate blocks that may leak information. The modified Blowfish is faster compared to Twofish with an encryption and decryption average time of 2418.08ms and 4002.70ms. The added derivation improved the avalanche of the modified blowfish. Blowfish achieved 47.14% while modified Blowfish attained 52.86%.

Corresponding Author:

Theda Flare G. Quilala,
Technological Institute of the Philippines,
938 Aurora Blvd., Cubao, Quezon City, Philippines,
Email: tfgquilala@gmail.com

1. INTRODUCTION

Security is involved with the protection of network and data while communicating over the public networks [1]. It is one of the prominent areas of concern in communication and data transmission, particularly in open networks, like the Internet. Examples of sensitive information transmitted through public communication facilities are financial transactions, medical and personal records [2]. As a result, various hackers always try to break into the system to steal critical information or to destroy the integrity of data [3]. With the increasing growth of science and study in the field of the network, we have the responsibility to secure our image, and data from third parties [4].

One way of guaranteeing the protection of information is through the application of cryptography. Cryptography is the practice and study of information hiding and achieving security by encoding messages to make them non-readable [5]. The use of cryptography addresses data privacy preservation and security from modification and unauthorized access during transmission [6]-[10]. Some popular and well respected symmetric-key block ciphers currently available include DES, 3DES, CAST5, RC6, CAST5, Blowfish, Twofish, Serpent, AES (Rijndael), TEA, IDEA, Serpent, and MARS [11], each exhibiting its strength and weakness.

Bruce Schneier designed Blowfish algorithm in 1994; it is a symmetric block cipher that aims to replace the outdated DES. Blowfish is a 64-bit variable length symmetric block cipher [12]. Blowfish is one of the fastest, compact, easy to understand, easy to implement, free alternative to existing encryption algorithms and features variable security level except when changing keys [13]. Several types of research were already conducted to test the security provided by Blowfish and results indicate that it is indeed fast and secure [14]-[16]. An attack developed by John Kelsey could break 3-round Blowfish; however, he was not

able to expand it. Disclosure of F allows performing a differential cryptanalysis which can recover all the rest of the key with 248 chosen plaintexts against the number of rounds reduced to eight [17] but was deemed impossible for round 8 and higher. To date, there is still no known attacks on the complete 16 rounds of blowfish.

Despite the fact that blowfish is a remarkably fast block cipher, extending it to act on 128-bit is the most natural manner [18]. In 1997, a request for candidate algorithm nominations for the Advanced Encryption Standard listed minimum functional requirements and asked for a symmetric block cipher capable of supporting block lengths of 128 bits and a key length of 128 bits [19]. Twofish, an encryption algorithm based on Blowfish, accepts 128-bit block size and was submitted and qualified as one of the finalists for AES [20] that provides a substantial level of security but lacks in encryption speed as compared to blowfish [21]. Twofish has seen less widespread usage than Blowfish [22].

Blowfish algorithm consists of two parts-key expansion and data encryption. In the key expansion, applying XOR to the variable length key and plaintext are used to produce the subkeys and generate the four key-dependent s-boxes. Each round requires around four kB which made the algorithm inapplicable for devices with a small memory like a smart card and phone. Using the algorithm, computation of the subkeys every time results in slower operation which made the algorithm inefficient to use in an application that requires changing secret key frequently [23]. However, three possible simplifications recommended by Schneier aimed at decreasing memory requirements and execution time. Those suggestions include the use of fewer and smaller S-boxes, fewer iterations from 16 to 8, and on-the-fly subkey calculation [12].

Other studies improved key generation of Blowfish to minimize the time required to produce subkeys [23]-[26]. For all studies involving the production of subkeys, the generation of elements achieved reduced time complexity though approaches used was entirely different from that of the original algorithm. Some researchers focused on the security aspect by modifying the f-function [27]-[29], but the latter concluded that the original blowfish algorithm was still more compact and more secure. Some optimizations on blowfish concentrated on the modification of the number of rounds to increase speed and ultimately enhance security [30] Although there was a mention of a minimum of five rounds, there was no minimum number set [31]. However, the recommended number of rounds is 16-8. Optimization on the reduction of S-boxes from four to two was also applied to increase the speed [32]. Several researchers have attempted to extend the block size of blowfish to 128-bit [18], [33] [34], but results indicate an increase in time complexity and need for higher memory.

From here, a modified Blowfish algorithm is proposed accepting block size of 128-bits and a key size of 128 bit to meet AES requirements that would exhibit speed and simplicity comparable to original blowfish. Iteration count and the number of S-boxes was reduced to achieve speed during key generation and replacement. For on-the-fly subkey calculation, there was an added S-box derivation technique.

This research aims to propose a modified blowfish algorithm that uses 128-bit block size and 128-bit key with a reduced number of iterations while maintaining the original structure of blowfish for a smooth migration. This study sought answers to the following objectives: to compare the execution of the modified Blowfish, Blowfish and Twofish algorithm regarding speed based on encryption, and decryption; and to compare the performance of the modified algorithm and blowfish regarding security using avalanche effect.

This study will modify Blowfish to make use of 128-bit block size and key, a design criterion set during the AES competition. The change in block size would allow encryption of file with lesser chances of having duplicate blocks. The original structure of Blowfish will still be used but will reduce the number of s-boxes from four to two to provide less memory consumption. The modification made on the algorithm ensures compatibility to the original version of Blowfish. A derivation technique will be introduced to reduce time in the generation of the key and to reduce symmetry in the s-boxes.

2. RESEARCH METHOD

2.1. Research Design

The key expansion will still convert the 128-bit key length into several subkey arrays. The modified Blowfish reduced the size from the previous 4168 bytes to 2128 bytes. These keys may also be generated separately and stored before any data encryption or decryption occurs. The P-array will now consist of 20 (P1, P2…P20) 32-bit subkeys. The four S-Boxes will still consist of 256 individual entries comprising 32-bits each (S1 - 0…255, S2 - 0…255). In the modified key expansion scheme, the total number of iterations will be reduced to 266 to generate all required subkeys.

Calculation of the subkeys are done using the same Blowfish algorithm, but the algorithm reduced the size to two S-boxes:

1. Initialization of the P-array followed by the four S-boxes was executed using constant strings that consist of the hexadecimal digits of pi.

2. P1 is XORed with the first 32 bits of the key, P2 is XORed with the second 32-bits, continuously until all bits of the key is exhausted up to P20. Repeat the cycle until the whole P-array has been XORed against the key bits.
3. The Blowfish algorithm is used to encrypt an all-zero string using the subkeys described in the previous steps (1 and 2).
4. The outcome of step 3 substituted P1 and P2.
5. Using the Blowfish algorithm, encrypt the output of step 3 using the revised subkeys.
6. Results obtained in step 5 replaced P3 and P4.

This process is continuously repeated replacing all entries of the P array, followed by the two S-boxes with the output of the continually varying Blowfish algorithm. Figure 2 shows the new process of encryption of the modified blowfish algorithm. The structure of the original blowfish algorithm is still adopted, but the modified Blowfish reduce the number of iterations to 8.

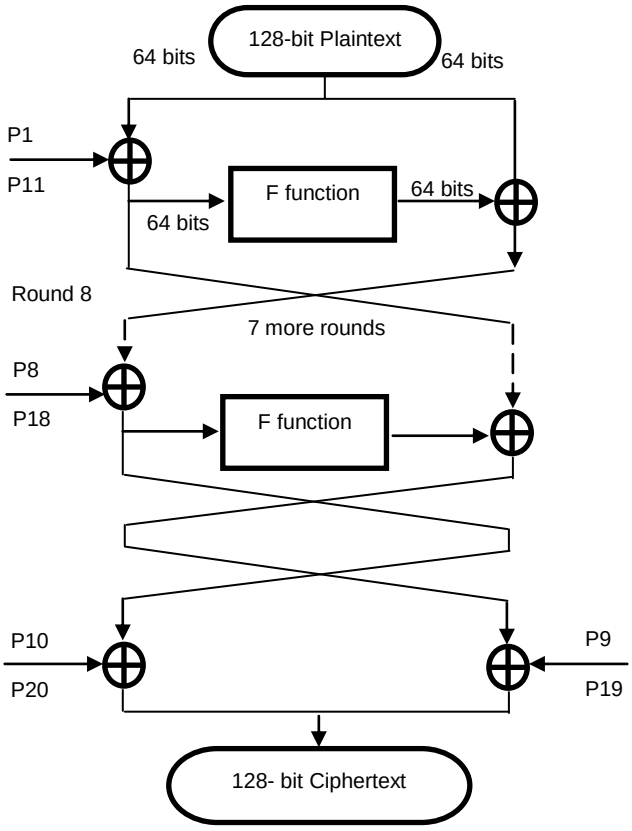

Figure 2. Blowfish proposed modification using 128-bit

The difference lies in the input block. The input block changes to 128-bit and will be split into two 64-bit equal segments LE0, RE0. Second, the first segment 64-bit block (LE0) is XORed to the first entry in the P-array (P1, P11) with two 32-bit entries. Third, input the two 32-bit data obtained to the F-function. The output from the F-function will then be XORed with the second segment (RE0) of the plaintext. Then, swap LE0 and RE0. This cycle will continue up to the eighth round. After the eighth round, exchange LE8 and RE8 reversing the last swap. Then, RE8 is XORed to P-array (P9, P19) and LE8 is XORed to P-array (P10, P20). Finally, we recombine LE9 and RE9 to get the ciphertext. The decryption process is the reverse of the encryption process.

Figure 3 also shows the details of the construction of the new F-function in the modified blowfish. The F-function now accepts a 64-bit data stream and will be divided into eight 8-bits where a is the first 8 bits, b is the second 8 bits, up to the last 8 bits. Transform each 8-bit data bits into a 32-bit data. The first four 8-bit data stream utilizes the first S-box while the next four 8-bit data stream uses the second S-box. The output from the S-boxes are then XORed or added to obtain the final 32-bit value per S-box and then concatenated to obtain the 64-bit output as shown in the equation:

$$F(LE0) = ((S1(a) + S1(b) << 1 \bmod 2^{32}) \oplus S1(c) >> 1) + S1(d<<1) \bmod 2^{32} \mid ((S2(e) + S2(f) << 1 \bmod 2^{32}) \oplus S2(g) >> 1) + S2(a<<1) \bmod 2^{32} \quad (1)$$

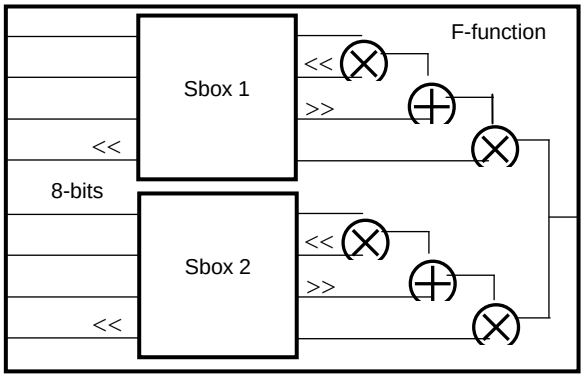

Figure 3. Modified F-function

The S-boxes are derived at runtime from S-box 1 by a simple rotation by one position of either the input or the output or either by left or right. Below defined the details of the derivation process:

$$S2(x) = S1(x) << 1 \quad (2)$$

$$S3(x) = S1(x) >> 1 \quad (3)$$

$$S4(x) = S1(x << 1) \quad (4)$$

The researcher changed the structure of the F-function as can be seen from the equation above.

2.2. Metrics

Performance metrics for the analysis of the performance of the algorithm is time (milliseconds), throughput (Mb/sec), and avalanche effect (%). Below is the description of the evaluation parameters:

1. Key Generation time: The amount of time needed to generate the subkeys
2. Encryption time: The amount of time necessary to change the plaintext to equivalent ciphertext.
3. Decryption time: The amount of time needed to change ciphertext to plaintext.
4. Throughput: Throughput indicates the speed of encryption. The size of plaintext divided by the total time is the calculation for throughput.
5. Avalanche effect: Refers to the characteristic where a minimal change in the input text results in a significant change of the output sometimes referred to as diffusion, reflecting the cryptographic strength of a cryptographic algorithm. Avalanche effect is calculated using hamming distance which is a measure of difference. It is the XOR calculation bit by bit of the ASCII value.

2.3. Research Procedure

Experimentation was done using different file sizes ranging from 10kb to 1000kb. The average time is computed using 20 trials of each file size. During the experimentation, the researcher used Intel® Core ™2 Quad CPU Q6600 @2.40 GHz with 4G RAM. The file and key used for all testing done were the same.

3. RESULTS AND ANALYSIS

As seen in Table 1, the average key generation time for Blowfish is 21.65ms while the modified Blowfish algorithm is 26.99ms. The number of rounds was reduced to 8 rounds to compensate for the time difference. Although the original algorithm is still faster than the modified version, the modified algorithm already uses 128-bit block size. Extending the block size to 128-bit lessen the chances of having duplicate blocks that may lead to the leak of information thus increases security. For a block cipher which uses 64-bit blocks, the threshold is about 32 gigabytes (232 blocks of 8 bytes). If a 1TB drive is encrypted, there exist 32 duplicated cipher blocks.

Table 1. Key generation time comparison in milliseconds using different file sizes

Input Size (kb)	BA	MBA
10	21.00	28.55
20	23.30	25.60
50	22.10	25.05
100	22.30	26.85
200	22.15	27.60
500	20.30	26.00
1000	21.00	27.70
Average (ms)	21.65	26.99

Comparing the key expansion of Blowfish and modified blowfish, Blowfish uses 4168 bytes while the modified Blowfish uses 2128 bytes. Bytes used for the modified blowfish is lower which would make it suitable for small devices with limited memory size. The difference is due to the differences in P-array and S-boxes used. Table 2 shows the comparison.

Table 2. Key generation comparison of bytes used

BA	MBA
Uses 4168 bytes, (computed as 4 bytes * 4 s-boxes * 256 entries each plus 4 bytes * 18 Parray entries)	Uses 2128 bytes (computed as 4 bytes * 2 s-boxes * 256 entries plus 4 bytes * 20 Parray entries)
The P-array consist of 18 (P1, P2…P18) 32-bit subkeys.	The P-array consist of 20 (P1, P2…P20) 32-bit subkeys
Four S-Boxes consists of 256 individual entries comprised of 32-bits each (S1 - 0…255, S2 - 0…255, S3 - 0…255, S4 - 0…255)	Two S-Boxes consists of 256 individual entries consisting of 32-bits each (S1 - 0…255, S2 - 0…255)
Number of iterations is 521 to generate all required subkeys	Number of iterations is 266 to make all the necessary subkeys

For the encryption and decryption time, as can be observed in Table 3 and Table 4, BA is still faster among the three algorithms having 1297.76ms and 2176.59ms average encryption and decryption time respectively and an encryption throughput value of 234.91Mb/sec and decryption throughput of 120.20Mb/sec. The difference in time is due to the difference in block size. The modified Blowfish is compared to Twofish, an encryption algorithm based on Blowfish because it also accepts 128-bit block size. Comparing modified blowfish and Twofish, each accepting 128-bit block size and key, the modified blowfish gained faster average encryption and decryption time at 1651.83ms and 2765.04ms compared to 2418.08ms and 4002.70ms for Twofish. Encryption throughput for modified Blowfish is at 182.92Mb/sec and decryption throughput is 89.65Mb/sec. For Twofish, encryption throughput is 124.32Mb/sec, and decryption throughput is 63.56. From the results, the modified Blowfish has higher throughput value meaning the modified algorithm has greater efficiency than Twofish.

Table 3. Encryption time comparison in milliseconds using different file sizes

Input Size (kb)	BA	MBA	TA
10	51.10	69.40	93.30
20	93.60	119.75	174.55
50	212.65	272.30	438.85
100	413.55	527.15	779.95
200	809.00	1030.45	1517.15
500	2007.55	2556.85	3737.90
1000	3999.40	5085.40	7423.25
Average	1297.76	1651.83	2418.08
Throughput	234.91	182.92	124.32

Table 4. Decryption time comparison in milliseconds using different file sizes

Input Size (kb)	BA	MBA	TA
10	91.80	153.75	179.85
20	174.00	255.30	336.45
50	402.90	562.05	833.45
100	827.00	1038.80	1538.50
200	1624.15	2047.55	3007.95
500	4045.35	5114.10	7413.50
1000	8070.90	10183.75	14709.20
Average	2176.59	2765.04	4002.70
Throughput	120.20	89.65	63.56

A suitable feature of every encryption algorithm is that a minimal change in the key should produce a significant difference in the ciphertext. The avalanche effect of the modified Blowfish is compared to Blowfish to ensure that the diffusion of the algorithm was not affected by the removal of the two S-boxes and to ensure that the derivation technique has removed the symmetry between the S-boxes.

The plaintext message used was "The quick brown fox jumps over the lazy dog" and varying 1 bit of the key. The following keys were used: FEDCBA9876543210, FEDCBA9876543211, FEDCBA9876543212, FEDCBA9876543213, and FEDCBA9876543214. Figure 4 shows the avalanche percentage. Blowfish has 47.14% avalanche, and the modified Blowfish is at 52.86%. The higher the avalanche percentage, the higher will be the security [35], this means that the modified algorithm even had a better avalanche, thus better security.

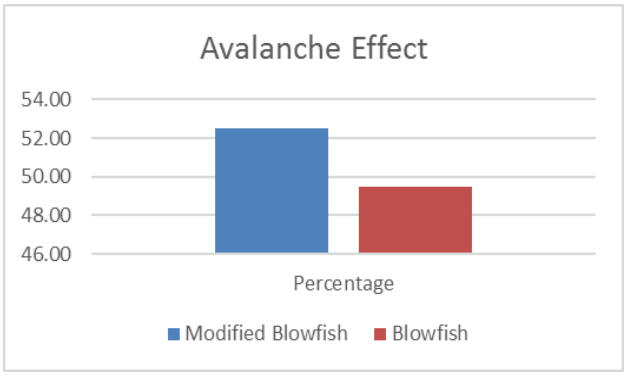

Figure 4. Avalanche effect of Blowfish and proposed modified blowfish

The efficiency of the modified algorithm was also evaluated using CrypTool 1.4.40 [36], a free and comprehensive e-learning program including cryptography and cryptanalysis. For this test, a file of 104Kb was encrypted using a 128-bit key. Blowfish accepts variable key sizes ranging from 32-448 bits. The modified Blowfish algorithm set the smallest key size to 128-bit. A 128-bit key has a complexity of 2^{128} or 3.40 e+38. For an encryption algorithm with a 128-bit key, a brute force attack will take 5e+025years which makes brute force utterly impossible. See Figure 5.

Entropy is a measure of randomness or uncertainty in the information usually referred to as confusion. Based from CrypTool, the document contains all 256 possible byte values. The entropy of the text is 7.99 with a maximum entropy value of 8.0. Encryption desires high randomness so that there is less or no dependency between key and ciphertext to make it difficult to guess by an attacker Frequency test is a statistical test. The CrypTool results indicate a passed result of 3.262253. Frequency test is a frequency analysis which studies the frequency of characters in a ciphertext.

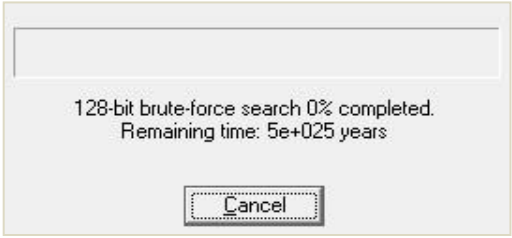

Figure 5. Brute Force on 128-bit key

4. CONCLUSION

This paper proposed a modified blowfish algorithm that uses 128-bit block size and a 128-bit key, maintaining the original structure of blowfish for a smooth migration with a reduced number of s-boxes to provide less memory consumption. Results show that the modified algorithm design continues to offer sufficient avalanche effect as the original with less storage requirement for the P-array and S-boxes.

Although the modified algorithm is a little slow compared to Blowfish, the increase in the input block size is the reason. Compared to Twofish, a related algorithm to Blowfish, the modified Blowfish is faster and has a better throughput efficiency. Based from Cyrptool, the modified Blowfish algorithm also passed the entropy and frequency test. In a 128-bit key, a brute force attack will take 5e+025years which makes brute force utterly impossible. Studies involving any data encryption or file encryption process can use the algorithm. For future works, other researchers may study hardware optimization implementation of the modified algorithm.

REFERENCES

[1] N. Kar, A. Majumder, A. Saha, A. Jamatia, K. Chakma, and M. C. Pal, "An improved data security using DNA sequencing," *3rd ACM MobiHoc Work. Pervasive Wirel. Heal. MobileHealth 2013*, pp. 13–18, 2013.

[2] G. Jacob and A. Murugan, "DNA based cryptography: An overview and analysis," *Int. J. Emerg. Sci.*, vol. 3, no. 1, pp. 36–42, 2013.

[3] A. Kaundal and A. Verma, "DNA Based Cryptography: A Review," *Ripublication.Com*, vol. 4, no. 7, pp. 693–698, 2014.

[4] N. Srividhya and T. Vino, "Genome based highly secured image using DNA cryptography and trellis algorithm," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 1658–1662.

[5] M. Bhattacharya, K. Pal, G. Ghosh, and S. S. Mandal, "Generation of novel encrypted code using cryptography for multiple level data security for Electronic Patient Record," *Proc. - 2015 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2015*, pp. 916–921, 2015.

[6] J. E. Camargo, D. F. Sierra, and Y. F. Torres, "Study of cryptographic algorithms to protect electronic medical records in mobile platforms," *Indian J. Sci. Technol.*, vol. 8, no. 21, pp. 1–7, 2015.

[7] S. Fong-In, S. Kiattisin, A. Leelasantitham, and W. San-Um, "A partial encryption scheme using absolute-value chaotic map for secure electronic health records," in *The 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE)*, 2014, pp. 1–5.

[8] A. R. Krishna, A. S. N. Chakravarthy, and A. S. C. S. Sastry, "A hybrid cryptographic system for secured device to device communication," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 6, pp. 2962–2970, 2016.

[9] Shivaputra, H. Sheshadri, and V. Lokesha, "A Naïve Visual Cryptographic Algorithm for the Transfer of Compressed Medical Images," *Bull. Electr. Eng. Informatics*, vol. 5, no. 3, pp. 347–365, 2016.

[10] M. Ranjan, A. H. Mondal, and M. Saikia, "A cloud based secure voting system using homomorphic encryption for android platform," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 6, pp. 2994–3000, 2016.

[11] M. Ebrahim, S. Khan, and U. Bin Khalid, "Symmetric Algorithm Survey: A Comparative Analysis," *Int. J. Comput. Appl.*, vol. 61, no. 20, pp. 975–8887, 2013.

[12] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (Blowfish)," in *Fast Software Encryption: Cambridge Security Workshop Cambridge, U. K., December 9--11,1993 Proceedings*, R. Anderson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 191–204.

[13] V. Kumar and A. Sharma, "A Survey on Various Cryptography Techniques," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 3, no. 4, pp. 307–312, 2014.

[14] A. Ramesh and A. Suruliandi, "Performance analysis of encryption algorithms for Information Security," in *2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT)*, 2013, pp. 840–844.

[15] G. Singh, A. Kr. Singla, and K. S. Sandha, "Superiority of Blowfish Algorithm in Wireless Networks," *Int. J. Comput. Appl.*, vol. 44, no. 11, pp. 23–26, Apr. 2012.

[16] P. Nema and M. A. Rizvi, "Critical Analysis of Various Symmetric Key Cryptographic Algorithms," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 3, no. 6, pp. 4301–4306, 2015.

[17] S. Vaudenay, "On the weak keys of blowfish," 1996, pp. 27–32.

[18] J. A. Mahdi, "Design and implementation of proposed BR encryption algorithm," *IJCCCSE*, vol. 9, no. 1, pp. 1–17, 2009.

[19] National Institute of Standards and Technology, "Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard," *Fed. Regist.*, vol. 62, no. 177, pp. 48051–48058, Oct. 1997.

[20] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, and C. Hall, "Twofish : A 128-Bit Block Cipher," *NIST AES Propos.*, vol. 15, no. 1, pp. 1–27, 1998.

[21] R. Bhanot and R. Hans, "A review and comparative analysis of various encryption algorithms," *Int. J. Secur. its Appl.*, vol. 9, no. 4, pp. 289–306, 2015.

[22] G. Muthukumar and E. G. Dharma, "A Comparative Analysis on Symmetric Key Encryption Algorithms," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 3, no. 2, pp. 379–383, 2014.

[23] T. S. Atia, "Development of a new algorithm for key and S-box generation in blowfish algorithm," *J. Eng. Sci. Technol.*, vol. 9, no. 4, pp. 432–442, 2014.

[24] J. C. S. B and G. S. Raman, "Ensemble of Blowfish with Chaos Based SBox Design for Text and Image Encryption," *Int. J. Netw. Secur. Its Appl.*, vol. 3, no. 4, pp. 165–173, 2011.

[25] A. A. Abd El-Sadek, T. A. El-Garf, and M. M. Fouad, "Speech encryption applying a modified Blowfish algorithm," in *2014 International Conference on Engineering and Technology (ICET)*, 2014, pp. 1–6.

[26] A. M. Alabaichi, "A Dynamic 3D S-Box based on Cylindrical Coordinate System for Blowfish Algorithm," *Indian*

J. Sci. Technol., vol. 8, no. 30, pp. 1–17, Nov. 2015.
[27] J. Raj and S. Ross, "Enhancement of Blowfish Encryption in Terms of Security Using Mixed Strategy Technique," *IIOAB J.*, vol. 7, no. 9, pp. 69–76, 2016.
[28] V. Poonia and N. S. Yadav, "Analysis of modified Blowfish algorithm in different cases with various parameters," in *2015 International Conference on Advanced Computing and Communication Systems*, 2015, pp. 1–5.
[29] S. Manku and K. Vasanth, "Blowfish encryption algorithm for information security," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 10, pp. 4717–4719, 2015.
[30] P. Patel, R. Patel, and N. Patel, "Integrated ECC and Blowfish for Smartphone Security," *Procedia Comput. Sci.*, vol. 78, no. December 2015, pp. 210–216, 2016.
[31] R. Patel and P. Kamboj, "Security Enhancement of Blowfish Block Cipher," 2016, pp. 231–238.
[32] L. Christina and J. I. V S, "Optimized Blowfish Encryption Technique," *Int. J. Innov. Res. Comput. Commun. Eng. (An ISO 3297 2007 Certif. Organ.*, vol. 2, no. 7, pp. 5009–5015, 2014.
[33] N. J. Oishi, A. Mahamud, and Asaduzzaman, "Short paper: enhancing Wi-Fi security using a hybrid algorithm of blowfish and RC6," in *2016 International Conference on Networking Systems and Security (NSysS)*, 2016, pp. 1–5.
[34] A. M. Alabaichi, R. Mahmood, F. Ahmad, and M. S. Mechee, "Randomness Analysis on Blowfish Block Cipher Using ECB and CBC Modes," *J. Appl. Sci.*, vol. 13, no. 6, pp. 768–789, Jun. 2013.
[35] B. S. Ross and V. Josephraj, "Performance Enhancement of Blowfish Encryption Using RK-Blowfish Technique," *Int. J. Appl. Eng. Res.*, vol. 12, no. 20, pp. 9236–9244, 2017.
[36] Bernhard Esslinger, "Cryptool Portal - Cryptography for everybody," 1998. [Online]. Available: https://www.cryptool.org/en/.