Computer Science and Information Technologies
Vol. 2, No. 1, March 2021, pp. 16~25
ISSN: 2722-3221, DOI: 10.11591/csit.v2i1.p16-25

An optimized encryption algorithm and F function with Dynamic substitution for creating S-box and P-box entries for Blowfish Algorithm

Rekha C, Krishnamurthy G N
Department of Computer Science and Engineering, BNM Institute of Technology, Bengaluru, Karnataka, India

Article Info

Article history:

Received Apr 26, 2020
Revised Jun 17, 2020
Accepted Jul 19, 2020

Keywords:

Blowfish
Correlation coefficient
Entropy
Floating frequency
P-box
S-box

ABSTRACT

In the field of cryptography, there has been a massive amount of enhancement in manipulating the plaintext which is unreadable, less prone to crackers and hackers, again manipulating this unreadable form to get back plaintext in some way. The Blowfish algorithm is a block cipher, has complex in structure in generating P-box and S-box entries using encryption algorithm. By simplifying the structure of encryption algorithm as well as F function with dynamic substitution, this can improve the performance by generating P-box and S-box entries of blowfish algorithm. In this paper, the proposed method simplifies the structure to produce P-box and S-box entries in order to reduce computational cost and demonstrates the performance of blowfish. The approach considers different security aspects namely EQ analysis, KS analysis, AV analysis, Entropy, Floating Frequency analysis and correlation of horizontally adjacent pixels in an encrypted image.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Rekha C,
Department of Computer Science and Engineering,
BNM Institute of Technology,
Bengaluru, Karnataka, India.
Email: rekhac1976@gmail.com

1. INTRODUCTION

Cryptography plays an important role in network security [1], that transfers sensitive information across insecure networks using encryption and decryption process. In cryptography the Key are confidentiality, integrity, and authentication [2], [3]. Cryptographic algorithm is categorised into two different types symmetric and asymmetric key cryptography. In symmetric key cryptography, only one key is used to encrypt and decrypt the information. The key plays a major role in symmetric key cryptography. Depending on the basis of operation symmetric key cryptography is divided into two types stream cipher and block cipher. A block cipher is the one where a block of plaintext is converted into ciphertext block of same length [4]. One example of symmetric block cipher is blowfish. A blowFish is a 16 round Feistel network which operates on plaintext with 64 bit blocks converted to ciphertext of 64 bit blocks, using a key which is ranging from 32 bits to 448 bits used in the encryption and decryption of plaintext. Blowfish algorithm includes two procedures: a key-expansion procedure and a data-encryption procedure. Data encryption function take place via 16 round Fiestel network as shown in Figure 1, each round having permutation and substitution, using F-function as shown in Figure 2, with key dependent. All operation performed are XORed and additions on 32-bit words and additional operations are four indexed array data lookups per round. Key expansion procedure converts 448-bit key into few subkey arrays of 4168 bytes [4], [5].

The key expansion procedure uses a key to generating P-box and S-box. Initialization of 18-P array using key taken as P0 from P-array is XORed with first 32 bits of the key. P1 XORed with second 32 bits of the key. Repeat this cycle until all P-array XORed with key bits. After initialization, pass two values of P value (P0 and P1 as L0 and R0) to the function Encrypt as mentioned in Figure 1, which generate two different encrypted key values L17 and R17. The output L17 and R17 of encryption function, is then copied to P0 and P1. Repeat this step until all 18-P value entries are generated continuously in order by replacing the output. In the same way S-box entries are initialized with fixed string like 'pi' value or zero, then pass two values of S-box (S0 and S1 as L0 and R0) to the encryption function as shown in Figure 1, generates two different encrypted values like L17 and R17 which is copied to S0 and S1. This step Continues till, replacing the output by changing continuously in order all entries of four S-boxes of Blowfish algorithm. The function F works takes 32 bit value as an input, and it divids into four 8 bit data. Each four 8 bit data is used for substitution. first 8 bit is used to get 32 bit value from S-box 0, second 8 bit data is used to get 32 bit value from S-box 1, third 8 bit data is used to get 32 bit value from S-box 2 and finally last 8 bit for S-box 3. Then first 32 bit value is added with second 32 bit value , the output is XORed with third 32 bit value, the output is then added with fourth value will get final 32 bit value as shown in Figure 2. To perform this initialization and generation of P-box and S-box takes more time because of 16-round of encryption algorithm.

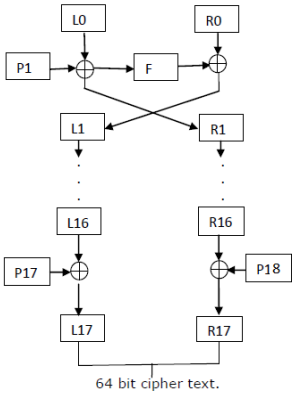

Figure 1. Encryption algorithm for blowfish

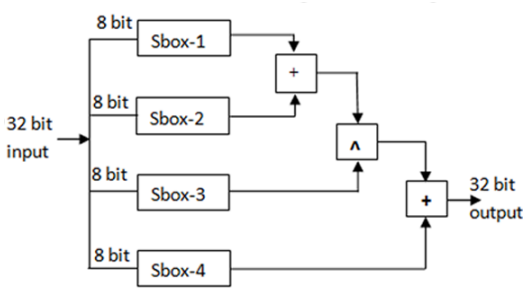

Figure 2. Internal operations of F function

The blowfish algorithm is fast and useful block cipher. many implementation has been conducted either through software or hardware. But very few implementation has been proposed to generate entries of P-box and S-box for blowfish algorithm. A new secret key as been proposed for block cipher, blowfish, which is a Feistel network with block size 64 bit and 32-bit to 448-bit a variable key [5]. The algorithm implemented with a complicated initialization and large data caches of 32-bit microprocessor. Implemented a novel VLSI architecture for blowfish algorithm which is based on partial pipelined structure [6]. The author in this paper has used two different techniques of modified Feistel function, first is iterative method and, second is partially pipelined technique. A four stage pipelined architecture is used along with two iterations that increases the area occupied with increasing throughput of the algorithm when compared with the two stage pipelining with 8 iterations that will reduce the area occupied with reduced throughput. Discusses general optimization principles of designing the algorithms, and performance analyzes of RC4, SEAL, RC5, Blowfish, and Khufu/Khafre on the Intel Pentium with respect to those general optimized principles [7]. Presented, a one round VLSI architecture of the BLOWFISH, which is based on the loop-folding technique combined with secure different modes (ECB, CBC2, CFB2 and OFB2) of operation [8]. The architecture uses a prototype chip to implement by using 0.35 /spl mu/ CMOS technology. Presented DRIL architecture, which is a four-tier architecture involving both software and hardware designs, for implementing blowfish algorithm using architectural features like inner loop pipelining and loop folding with dynamic reconfiguration [9]. The main objective of the research which is presented in this paper is to develop an algorithm with low-power, high throughput, real-time, reliable and secure crypto system, that can be achieved through hardware implementations [10]. Implement a new secret-key 'Blow-CAST-Fish' block cipher that uses good features of both CAST-128 and Blowfish algorithms using VHDL implementation [11]. Proposed a modified the Blowfish algorithm by enhancing its performance in terms of speed, Throughput, Power consumption and Avalanche effect [12]. Author has proposed a way to enhance the performance of the Blowfish cryptography algorithm by introducing parallel processing technique and making modifications to the Fiestel (F) function of Blowfish by combining

the Blowfish and the Runge-kutta (RK) Method. The F function of Blowfish has been modified with different formulae and the outcome of a series of RK-Blowfish algorithms were compared with the Blowfish algorithm.

The blowfish algorithm has been widely used in network security method to enhance the security by implementing through software or hardware based on variety of aspects like speed, security, portability etc. The scope of this work includes generation of P-box and S-box entries using a modified fiestel network, which is simple structure, in order to reduce the computational cost of generating P-box and S-box entries in blowfish algorithm. The main motivation of proposed work is to design and implementation of generating P-box and S-box entries by reducing the number of rounds instead of 16- rounds of encryption algorithm to overcome the limitation of blowfish algorithm. In this paper, a simple P-box and S-box generating algorithm to overcome the computational cost is designed and implemented using modified fiestel network.

2. THE PROPOSED APPROACH

In this section, an approach for generating P-box and S-box for blowfish algorithm is presented. The blowfish algorithm takes P-array values, initialized by master key K, S-box, initialized by Pi or zero value, will be generated through modified encryption algorithm procedure. The encryption procedure modified by reducing number of rounds, 9 iterations with 9-rounds, instead of 9 iteration with 16-rounds in the procedure. By reducing the number of rounds in the encryption procedure we can reduce the time as well reduction in the computational cost of blowfish algorithm. The algorithm takes 16-byte Key K (K0 K1 K2 K3 K4 K5 K6 K7 K8 K9 K10 K11 K12 K13 K14 K15 K16), as an input to generate all 18 P-values. First 4 word K0K1K2K3 from key is stored in P0 and seconde 4 word K4K5K6K7 from key is taken in P1 similarly all key values are stored in P-values like

[K0 K1 K2 K3] = P(0)
[K4 K5 K6 K7] = P(1)
[K8 K9 K10 K11] = P(2)
[K12 K13 K14 K15] = P(3)
[K0 K1 K2 K3] = P(4)..... P(18)

After initialization, pass two values P0 and P1 each of 32-bit, (as left half L0 and right half R0) to modified algorithm. In the algorithm,

L1 = L0 is XORed with P0 and
R1 = (R0 is XORed with P1) XORed with F-function with input L1 and S-box

Then swap L1 and R1 and considered as input for next iteration. Repeat these steps to get the output with two values P0 and P1, for this the modified encryption algorithm takes 9 iterations instead of 16 iterations. The flowchart of the proposed modified encryption algorithm is given in Figure 3 and the algorithm for proposed encryption algorithm 1 is given in section 4.1. The F function takes 32 bit value as an input and divids into four 8 bit value. Each four 8 bit value is used for substitution from each four S-box and first 4 bit from each 8 bit value is considered from which S-box we should get the value. The flowchart of the proposed modified F- function is given in Figure 4 and the algorithm 2 is given in section 2.1.

Algorithm 1 Enhanced encryption algorithm for blowfish

```
Procedure ENCRYPTION ALGORYTHM
P0 amd P1 values are input for procedure as 32-bit L0 and R0
     for i = 0 to N + 2 do increment i = i + 2
L1 = L0 ⊕P_i  R1 = R0⊕P_{i+1}X = F(S − Box, L1)R1 = R1 ⊕ X
swap L1 and R1
     end for
The output values are L17 and R17 are copied to P0 and P1.
Repeat these steps to generate all entries of 18 P-values
end procedure
```

Algorithm 2 Modified F function for blowfish

1: **procedure** F FUNTION ALGORITHM
2: P0 and P1 values are input for this procedure as 32-bit L0 and R0, $a = x[3]$
3: $b = x[3]\&0x000f$
4: $c = x[2]$
5: $d = x[2]\&0x000f$
6: $e = x[1]$
7: $f = x[1]\&0x000f$
8: $g = x[0]$
9: $h = x[0]\&0x000f$
10: $y = S[b\ mod\ 4][a] \oplus S[d\ mod][c]$
11: $y = y + S[f\ mod\ 4][e]$
12: $y = y \oplus$ S[h mod 4][g]
13: returny
14: **end procedure**

2.1. Proposed modified encryption algorithm and dynamic substitution in F-function

The main goal of this work is to provide more security, minimizes the time taken for generating P-box and S-box entries and computational cost. The proposed algorithm includes reduction of computational cost by reducing the number of iterations in the encryption algorithm with 9 rounds and 9 iterations, where original algorithm uses 16 round and 9 iterations in order to generate P-box and S-box entries. And also provides better security by using dynamic substitution in modified F function.

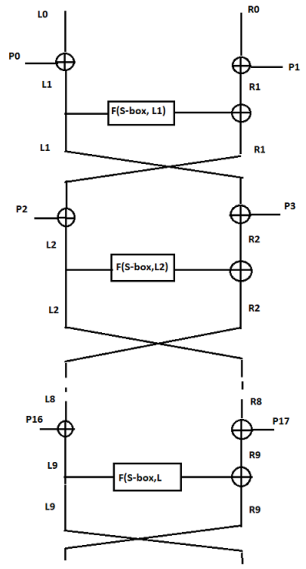

Figure 3. Flowchart of the modified encryption algorithm

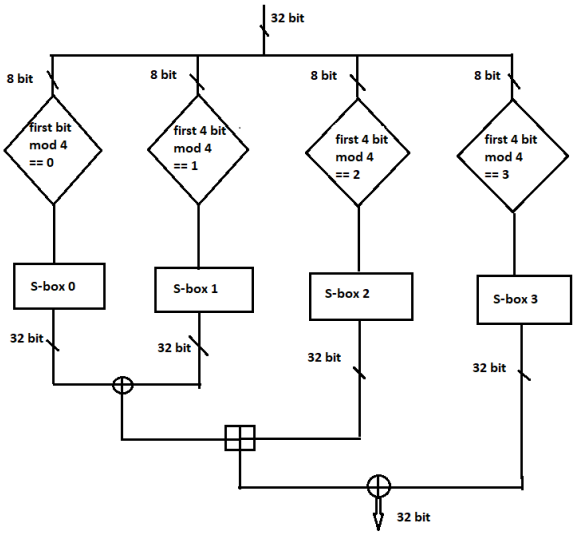

Figure 4. Flowchart of the modified Function F

3. RESULTS AND DISCUSSION

In this section, The original blowfish algorithm [5] and modified blowfish algorithm are applied on the image Arms.bmp with the same key. The comparison are made on both original and modified algorithms by making use of avalanche effect, encryption quality, key sensitivity and statistical analysis. The original image of Arms.bmp in Figure 5, is encrypted and decrypted by applying original blowfish algorithm are shown in. Same original image is encrypted and decrypted using modified blowfish algorithm. The Figure 6 and Figure 7 shows the result of encrypted and decrypted image using original algorithm and Figure 8 and Figure 9 shows the result of encryption and decryption images by applying modified algorithm.

Figure 5. Original image armsbmp

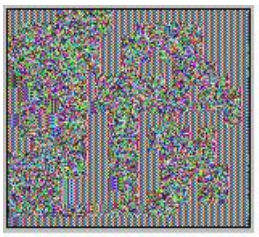

Figure 6. Encrypted image of arms bmp using original algorithm

Figure 7. Decrypted image of arms bmp using original algorithm

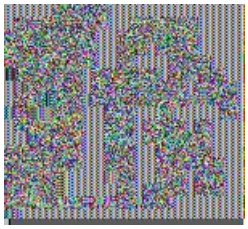

Figure 8. Encrypted image of arms bmp using modified algorithm

Figure 9. Decrypted image of arms bmp using modified algorithm

3.1. Encryption quality test

The encryption quality (EQ) test in \cite{13swf}, \cite{15swf} measures the quality of encryption which is based on the deviation between the plaintext image and ciphertext image. The more deviation of ciphertext compared to plaintext, better is the encryption algorithm. EQ test is the average number of changes to each grey level L between original and encrypted images. The mathematical formula and the steps to perform EQ test is given as:

$$EQ = \frac{\sum_{L}^{255}\left[H_L(F') - [H_L(F)]\right]}{256}$$

Let F and F' is represented as original image and encrypted image of size M*N pixels with L grey levels. At position (x,y), the grey levels of the F and F',($0 \le x \le M-1$, $0 \le y \le N-1$) is represented as F(x,y) and \(F'\)$(x,y) \in L$ ranging from $0 to 255$.

The steps are:

- Compute HL(F) the number of occurrences of each gray level L in the original image and HL(\(F'\)) denotes the number of occurrences of each grey level in the encrypted image.
- Compute the average number of changes to each grey level L using above given mathematical formula.

Encryption Quality test is calculated using both original and modified blowfish algorithm is shown in Table1.

Table 1. Comparison of encryption qualities of original and modified blowfish algorithm for different rounds

Number of Rounds	Original Blowfish	Modified Blowfish
2	190.351	167.570
4	201.898	202.28
6	200.132	196.507
8	203.484	180.179
10	202.796	203.601
12	203.273	199.390
14	202.046	202.859
16	192.414	202.781

3.2. Key sensitivity test

A key of 16-character with 128 bits is used for encryption and decryption. The key sensitive test [5], [14] has been carried out as follows

- Applying the 16-character 128-bit key, Key1, to Encrypt an image Arms.bmp by original algorithm.
- Change any randomly selected one bit from Key1. Then from this modified key, Key2, encrypt the same image by applying to original algorithm. Ex : ADF378 465E262AB1F5DEC94A0AF25**F**27, from this key1 we have randomly selected **F** as shown in bold and changed to **B** as ADF378 465E262AB1F5DEC94A0A25**B**27 which is key2.
- Apply these two keys to modified blowfish algorithm using the same image and then compared.
- The result is shown in below Table 2, comparison of both ciphered images which are encrypted by original as well as modified algorithm using these two different keys
- The result is 99.384781 of difference, when we encrypt the image using original blowfish algorithm with key1 and when we encrypt the same image using same original algorithm with key2, in terms of grey scale values where there is only one bit difference in these two key2.
- The result is 99.539299 of difference when we encrypt the image using modified blowfish algorithm with key1 and when we encrypt the same image using same algorithm with key2.

Table 2. Comparison of key sensitivity of original and modified blowfish algorithm for different rounds

Number of Rounds	Original Blowfish	Modified Blowfish
2	95.86	96.049
4	99.567	99.578
6	99.459	99.583
8	99.612	99.652
10	99.590	99.602
12	99.599	99.588
14	95.588	99.580
16	99.410	99.628

3.3. Avalanche effect

The avalanche effect in [5], [13-14] means if there is a change in one bit in the plain text then there will be number of bits changes in the cipher text. To compute avalanche effect we need to change one bit from the plain text (image Arms.bmp), named as an image BArms.bmp, and then encrypt this image using both original blowfish and modified blowfish algorithms. Here the proposed algorithm is compared with original algorithm at different rounds along with four cases. The Table 3 provides which algorithm gives better avalanche effect.

Case 1: Comparing Avalanche effect for encrypted image of Arms.bmp and B1Arms.bmp with same key1 using original and modified algorithm.

Case 2: Comparing Avalanche effect for encrypted image of Arms.bmp and B1Arms.bmp with same key2 using original and modified algorithm.

Case 3: Comparing Avalanche effect for encrypted image of Arms.bmp with key1 and key2 using original and modified algorithm.

Case 4: Comparing Avalanche effect for encrypted image B1MAND.bmp with key1and key2 using original and modified algorithm.

Table 3. Comparison of avalanche effect of original and modified blowfish algorithm for different rounds with four cases

Number of Rounds	Case 1		Case 2		Case 3		Case 4	
	Org	Mod	Org	Mod	Org	Mod	Org	Mod
2	1123	2527	770	1588	2015	1612	1630	2013
4	1149	2390	761	1547	2049	1620	389	3328
6	1137	2416	772	810	1131	1652	2473	1155
8	992	2552	716	1570	1131	1652	1162	2440
10	2340	1186	1513	778	2437	1151	1976	1623
12	2335	1192	882	1428	1168	2851	1069	2471
14	2379	1182	825	1474	1953	1646	2425	1152
16	1206	2333	800	804	1999	1141	2439	1135

3.4. Correlation coefficient

The correlation coefficient is determined relationship between horizontally adjacent pixels in an image [9], [11]. The steps for determining the correlation of horizontal adjacent pixels in an image Arms.bmp is as follows

- From the original image and their encrypted image, select N pairs of horizontally adjacent pixels.
- Select pixels randomly and pixels adjacent to them from the both original image (Arms.bmp) and their encrypted images using both original algorithm as well as modified algorithm.
- Using r_{xy} formulae to find correlation coefficient, where x and y represent grey scale values of horizontally adjacent pixels in an image.

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

where D(X) and D(Y) represents the variance of x and y values and COV(X,Y) is covariance of x and y and is given by

$$cov(x,y) = \frac{1}{N}\sum_{i=0}^{N}\big(x_i - E(x)\big)\big(y_i - E(y)\big)$$

Where N represents number of horizontal adjacent pixels selected randomly, E(X) and E(Y) represents the mean values of x and y values. This test is carried out for about randomly selected horizontally adjacent pixels from the original image Arms.BMP and encrypted images. Then using above equations correlation coefficient will be computed and is as shown in below Figure. 10, Figure. 11, and Figure. 12. The correlation coefficient of original image is 0.072053 and for cipher image which is encrypted by blowfish algorithm is 0.005616 and for modified algorithm is -0.429036. In both original and modified algorithm the correlation coefficients for plaintext image with that of ciphertext images are far apart.

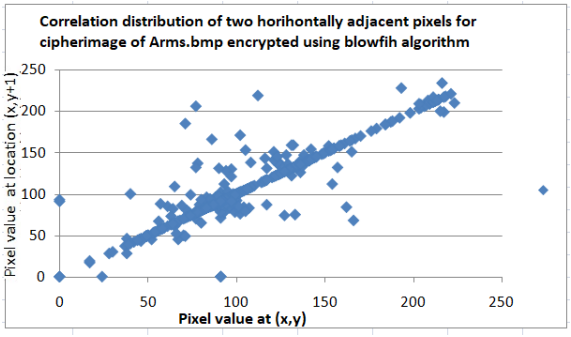

Figure 10. Correlation distribution of two horizontally adjacent pixels for original image arms BMP

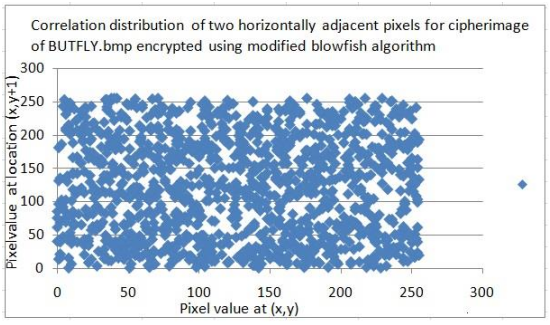

Figure 11. Correlation distribution of two horizontally adjacent pixels for encrypted image using original algorithm

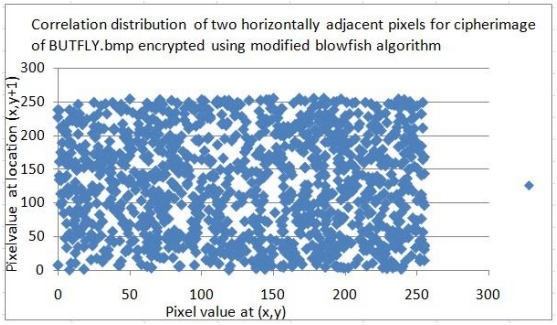

Figure 12. Correlation distribution of two horizontally adjacent pixels for encrypted image using modified algorithm

3.5. Entropy

The entropy [15] of an information data is measured in bits per character.The entropy is calculated as the average amount at which information data is produced by a truly random source of data. To calculate this mean, the individual information are weighted with the probabilities of their occurrence. The mathematical formula for calculating entropy is

$$H(m) = \sum_{i=0}^{2N-1} p(m) \log_2 \frac{1}{p(m_i)} bits =$$

where p(m_i) be the probability of occurrence of a character and entropy is presented in bits. After evaluating the above equation 4, we obtain its entropy as H(m) = 8 , which is corresponding to a truly random source. Given an information source that generates random messages, in general its entropy value is lesser than the ideal one. However, when the messages are encrypted, their entropy should be 8. If the output of a block cipher emits with entropy less than 8, there exists certain degree of predictability, which endanger its security. Let us consider the ciphertext of an image, encrypted using the original as well as modified algorithm, the number of occurrence of each ciphertext block and the probability of occurrence is computed. The obtained information entropy is very much close to the theoretical value of 8. This means that leakage of the information data in the encryption process is negligible and the encryption system is secure against entropy attack. The entropy is calculated using equation. 4 and is listed in below Table 4.

Table 4. Comparison of entropy for different rounds of original and modified blowfish algorithm

Number of Rounds	Original Blowfish	Modified Blowfish
2	6.74	6.86
4	7.03	7.03
6	6.96	7.07
8	7.06	7.05
10	7.07	7.07
12	7.06	7.06
14	7.05	7.06
16	7.07	7.07

3.6. Floating frequency

The floating frequency [16] how many different characters are to be found in any given 64-character long segment in ciphertext. Frequency analysis is based on certain letters and combinations of letters occur with varying frequencies. The frequency function considers sequences of characters in the 64 characters long segment and counts how many different characters are to be found in this 64-character long segment. Then the segment is shifted one character to the right and the calculation is repeated. This procedure results in a summary of the ciphertext which identify the places with high and low information density. Depending on the structure and content of the data in the segment, encrypted images (bmp files) values obtained usually lie between 5 and 20, as shown in Figure 13 and the floating frequency for the encrypted images using original and modified algorithm is aa shown in Figures 14 and 15.

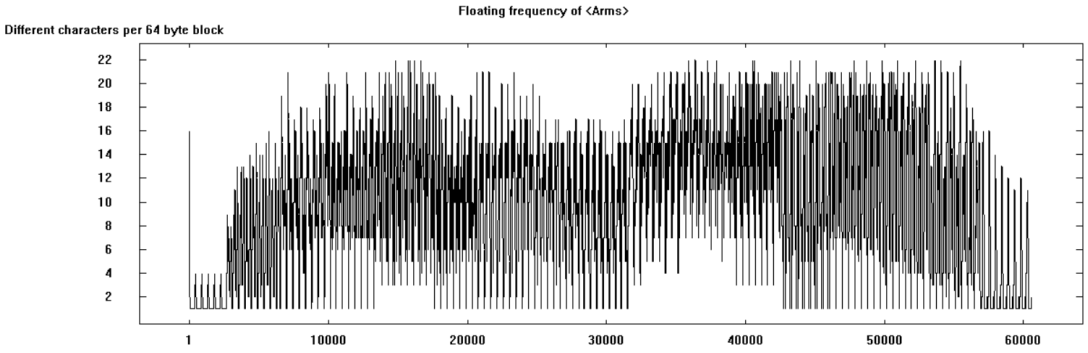

Figure 13. Floating frequency for original image arms bmp

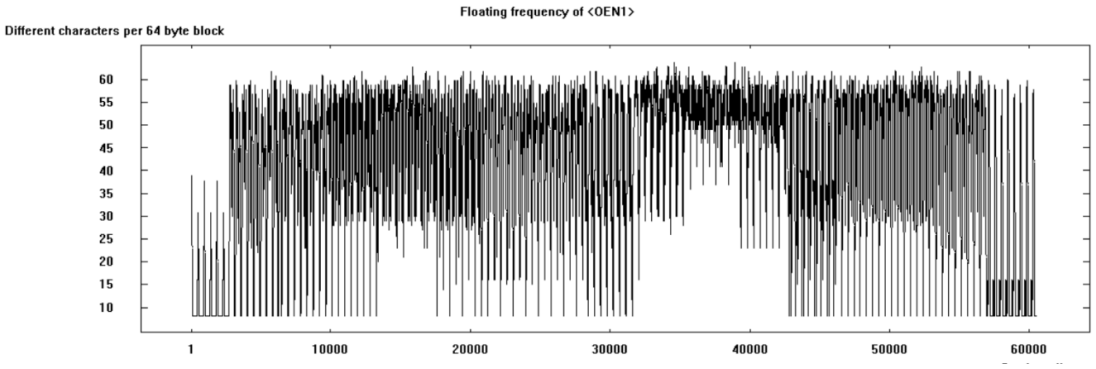

Figure 14. Floating frequency for encrypted image using original algorithm

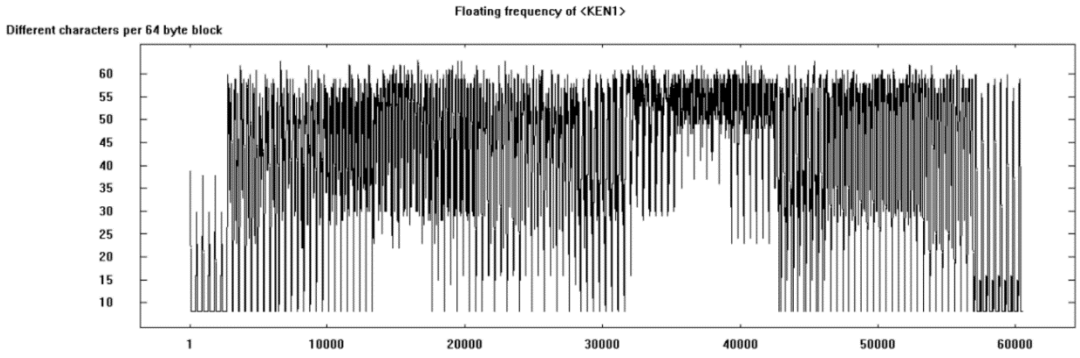

Figure 15. Floating frequency for encrypted image using modified algorithm

4. CONCLUSION

To enhance the security features of blowfish algorithm, the proposed method has been designed and implemented to create S-box and P-box values of blowfish algorithm using modified encryption algorithm and modified F function with dynamic substitution. The main motivation behind for proposed algorithm is to reduce the time for generating s-box and P-box values by reducing the number of rounds, 9 iterations with 9 rounds, instead of 9 iterations with 16 rounds in the encryption algorithm for blowfish algorithm. From the results, the proposed modified encryption algorithm performs better in all the aspects when compared with the original blowfish algorithm.

REFERENCES

[1] M. A. Kumar, S. Karthikeyan. "Investigating the Efficiency of Blowfish and Rejindael (AES) Algorithms," *International Journal of Computer Network and Information Security*, vol. 4, pp. 22-28, 2012, doi: 10.5815/ijcnis.2012.02.04.

[2] Behrouz A. Forouzan. "Cryptography and Network Security", *Tata McGraw-Hill, 2nd edition*, 2008.

[3] M. A. Kumar and S. Karthikeyan. "Investigating the Efficiency of Blowfish and Rejindael Algorithms," *International Journal of Computer Network and Information Security*, pp. 22-28, 2012, doi:10.5815/ijcnis.2012.02.04.

[4] William Stallings. "Cryptography and Network Security," *Fifth Edition, Pearson Publication*, Prentice Hall, 2013.

[5] B. Schneier. "Description o f a new variable-length key, 64-bit block cipher (blowfish)," *International Workshop on Fast Software Encryption*, Springer, Berlin, Heidelberg, vol 809, pp 191-204, 1994.

[6] P. Karthigaikumar and K. Baskaran. "Partially Pipelined VLSI Implementation of Blowfish Encryption/Decryption Algorithm," *Int. J. Image Graph*. vol. 10, pp. 327-341, 2010, doi: 10.1142/S0219467810003809.

[7] Schneier B., Whiting D. "Fast software encryption: Designing encryption algorithms for optimal software speed on the Intel Pentium processor," In: *Biham E. (Eds) Fast Software Encryption. FSE 1997. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, vol 1267, 242-259, 1997.

[8] Yeong-Kang Lai and Yu-Chuan Shu, "A novel VLSI architecture for a variable-length key, 64-bit Blowfish block cipher," 1*999 IEEE Workshop on Signal Processing Systems. SiPS 99. Design and Implementation (Cat. No.99TH8461)*, 1999, pp. 568-577, doi: 10.1109/SIPS.1999.822363.

[9] Sudarshan T.S.B., Mir R.A., Vijayalakshmi S. "DRIL–A Flexible Architecture for Blowfish Encryption Using Dynamic Reconfiguration, Replication, Inner-Loop Pipelining, Loop Folding Techniques," In: Srikanthan T., Xue J., Chang CH. (Eds) *Advances in Computer Systems Architecture. ACSAC 2005. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, vol 3740, pp. 625-639, 2005.

[10] P. Karthigai Kumara and K. Baskaran. "An ASIC implementation of low power and high throughput blowfish crypto algorithm," *Microelectronics Journal*, vol. 41, no. 6, pp. 347-355, June 2010, doi: 10.1016/j.mejo.2010.04.004.

[11] Krishnamurthy G.N, Dr. V. Ramaswamy, Leela G.H and Ashalatha M.E. "Blow-CAST-Fish: A New 64-bit Block Cipher," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no.4, pp. 282-290, April 2008.

[12] G.N. Krishnamurthy, V. Ramaswamy and G.H. Leela, "Performance enhancement of Blowfish algorithm by modifying its function," *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications*, pp. 241-244, 2007, doi: 10.1007/978-1-4020-6266-7_44.

[13] V. Ramaswamy and G. N. Krishnamurthy, "Encryption quality analysis and security evaluation of blow-castfish using digital images," *Communicated to International Journal of Computational Science*, 2008.

[14] S. Shailaja and G N Krishnamurthy, "Comparison of Blowfish and Cast-128 Algorithms Using Encryption Quality, Key Sensitivity and Correlation Coefficient Analysis," *American Journal of Engineering Research (AJER)*, vol. 3, no. 7, pp-161-166, 2014.

[15] Yue Wua, Joseph P. Noonana, Sos Agaianb. "Shannon Entropy based Randomness Measurement and Test for Image Encryption," *Information Sciences*, 2011, doi: 10.1016/j.ins.2012.07.049.

[16] S. Kulshreshtha, V. Verma and R. Kalra. "Analytical View of Cryptographic Techniques through Cryptool," *Journal of Telecommunications*, vol. 10, no. 2, pp. 22-26, 2011.