# A Modified Approach of Blowfish Algorithm Based On S-Box Permutation using Shuffle Algorithm

Reynaldo R. Corpuz
Technological Institute of the Philippines
938 Aurora Blvd., Cubao, Quezon City1109 Philippines
xeonlime10@gmail.com

Bobby D. Gerardo
West Visayas State University
Lapaz, Iloilo City, Philippines
bgerardo@wvsu.edu.ph

Ruji P. Medina
Technological Institute of the Philippines
938 Aurora Blvd., Cubao, Quezon City 1109, Philippines
ruji.medina@tip.edu.ph

## ABSTRACT

Blowfish Algorithm has been widely analyzed and gradually accepted as a good and powerful cryptographic algorithm offering several advantages among which is suitability and efficiency for implementing hardware and software; it satisfies the basic requirements in cryptography with high in attack immunity and relatively low in algorithm complexity. In spite of undeniable success in cryptography, Blowfish (BF) still has drawbacks in the substitution and permutation of values in the substitution box (S-box) that can affect the encryption and decryption of the algorithm. This study adopted Fisher-Yates Shuffle (FYS) also known as Knuth shuffle (KS) for the permutation of S-box, and a modified Function F was used to enhance the BF algorithm to address this issue. The results show that Modified approach in Blowfish Algorithm outperformed the Original Blowfish in the encryption, decryption, and throughput of the algorithm.

## CCS Concepts

**•Security and privacy → Information-theoretic techniques**

## Keywords

Shuffle algorithm, Fisher-Yates Shuffle, Blowfish Algorithm, S—box, Function F, Substitution-permutation networks.

## 1. INTRODUCTION

A science and art of keeping message secure is cryptography, performed by cryptographers. It plays a vital role in information security against known attacks and decreases the risk of hacking information. The unprecedented use of digital communications, electronic financial transactions, and digital transactions have brought remarkable potential on security issues, and attacks against cryptography, hackers are undeniably much-sophisticated year-after-year[1]–[3]. In this context, cryptographic development has been of high priority and challenging research area in both fields of engineering and mathematics. Blowfish (BF) is a symmetric key block cipher algorithm which uses the same key, that can effectively use for both encryption and decryption. Bruce Schneier in 1993 designed it, BF is a variable length key, 64-bit block cipher, converts a key from 32 bits and at most 448 bits as input and 64-bit block as output[4], [5].

Blowfish Algorithm (BA) has been widely analyzed and gradually accepted as a good and powerful cryptographic algorithm offering several advantages among which is its suitability and efficiency for implementing hardware and software; it satisfies the basic requirements in cryptography with high in attack immunity and relative low in algorithm complexity[6]–[8]. The algorithm is available to the public domain, unpatented and can be used by anyone for free. Blowfish (BF) algorithm displayed undeniable success in cryptography. However, several papers mentioned some of its drawbacks[2], [6], [9]–[11]. These problems point out to the substitution and permutation of values in substitution box (S-box) which directly influenced in the overall encryption process. Improving the security of cryptosystem based on block cipher depends on the replacement of Substitution boxes in the encryption/decryption algorithm[12]. The generation of S-box depends on Pseudo-Random-Number-Generators and shared-secret-key. According to shared-secret-key all Pseudo-Random-Numbers are scrambled. After scrambling, the S-box is generated[13]. An efficient cryptographic S-box should have the characteristics of balanced high nonlinearity scores, low maximum differential approximation probabilities, an avalanche effect close to ideal value.

This paper details the Modified Approach of BF utilizing shuffle algorithm (SA) called Fisher-Yates Shuffle (FYS) also known as Knuth shuffle (KS) for permutation. To my knowledge and study, no one has yet utilized the FYS in the block ciphers: a comparison of its performance with the original BF regarding throughput, encryption time and decryption time using the different file formats to show how efficient the Modified approach of BF.

## 2. RELATED LITERATURE

Symmetric block ciphers security of data relies on the Substitutions box/es. S-box is the essential structures used to achieve resistant against different attacks, and robust block cipher encryption algorithms[13]–[16]. Substitution is a nonlinear transformation which performs confusion of bits. It provides the cryptosystem with the confusion property[16]. The earliest block ciphers were simple networks that combined substitution and permutation circuits and called substitution-permutation networks (SPN). In modern encryption algorithm, a nonlinear transformation is essential and is proved to be a strong cryptographic primitive against linear and differential cryptanalysis. Different methods have been presented to construct S-box against processing speed, security, and randomness of values[11]–[17]. The SA is applied to transform the equation

*ICNCC 2018,* December 14–16, 2018, Taipei City, Taiwan
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6553-6/18/12…$15.00
DOI: http://dx.doi.org/ 10.1145/3301326.3301331

positivity of a descriptor continuous-time and discrete-time linear systems by the use of SA has been addressed[18]–[20].

## 2.1 Blowfish Algorithm

BF Algorithm is a symmetric block cipher. A 64-bit data block cipher and a variable-length secret-key range from any length to 448 bits. The algorithm has fragments of two: for key-expansion and data encryption. The expansion key transforms a key of at most 448 bits into multiple subkey arrays with a total of 4168 bytes. Encryption of data occurs thru a 16 -round Feistel Network. For every round, it has a key-dependent permutation and a key-and-data-dependent switch. A 32-bit words additions and XORs for all operations Data lookups per round for the four indexed arrays are the only additional operations[4].

### *2.1.1 Expansion of Keys*

The BF algorithm uses a large number of subkeys. 18-P-arrays for 32-bit and 256 entries for each 32-bit S-boxes. All subkeys should perform before the encryption or decryption of data. Below is the exact process of BF:

1: With the use of hexadecimal digits of pi for the P-array initialization and then the four S-boxes

2: With the first 32-bit of the key will XOR to P1, then Second 32-bit of the key will XOR to P2 and so on for all the bits of the key until P18.

3: using the keys described in steps one and two all string zero must encrypt with the BA.

4: the output of step 3 replaces the values of P1 and P2.

5: using the Blowfish algorithm with the modified keys encrypt the output of step three.

6: output from step 5 replace P3 and P4.

7: the process continues, replacing all the elements of the P-array, then all S-boxes one after the other, with the continuously changing output of the BA.

### *2.1.2 Encryption of Data*

As stated previously, as shown in Figure 1 blowfish algorithm is a Feistel network that consists of 16 rounds. It has an input of 64-bit plaintext and subkeys (32 bits) of 18 P-array. An output of 64 bits ciphertext. In algorithm 1 describing the Blowfish Algorithm.

```
Input: Plain text (64-bit) and from P1 to P18
Output: Ciphertext (64-bit)
Divide plain text into two, and each has 32-bit for XLeft and XRight;
for i=1 to 16 do
    perform XLeft = XLeft XOR Pi;
    perform XRight = F(XLeft) XOR XRight;
    Swap XLeft and XRight (the last swap, undo.);
end
perform XRight = XRight XOR P17;
perform XLeft = XLeft XOR P18;
Recombine XLeft and XRight (ciphertext);
```

**Algorithm 1:** Blowfish Algorithm

In equation 1 of Function F is calculated in sequence order. A 32 bits XL is divided into four 8-bit quarters: a, b, c, and d. S boxes input are the 8-bit quarters. Outputs are added with modulo $2^{32}$ and XORed to produce the final 32 bits. Figure 2 showing the Function

F.

$$F(XL) = ((S_{1,a} + S_{2,b} \bmod 2^{32}) XOR\ S_{3,c}) + S_{4,d} \bmod 2^{32}$$

**Equation 1:** Function F

Decryption is the same in the encryption process, except for P1-P8 are used in the reverse order.

## 2.2 The BF Cryptanalysis

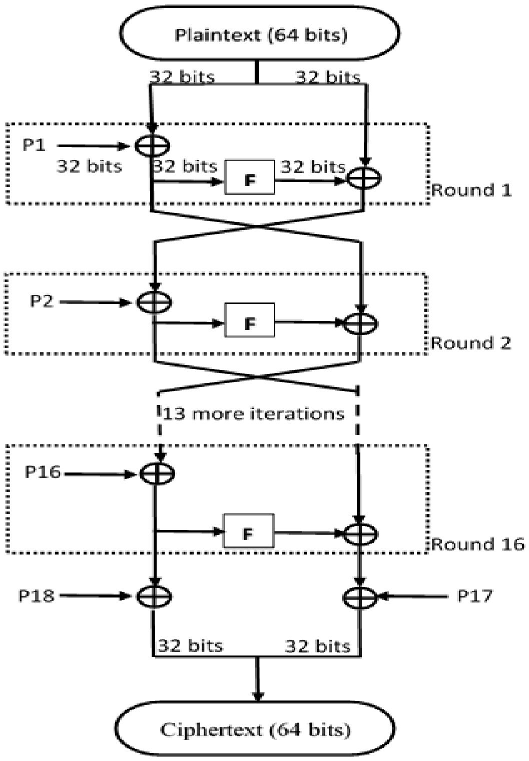


**Figure 1. Block Diagram of Blowfish**

Cryptanalysis is an attempt to unauthorized deciphering the plaintext that corresponds to the ciphertext. To date, blowfish algorithms full-round version is invulnerable against cryptanalysis. There are numerous schemes to break the cryptographic system, but none succeeded to extract information.

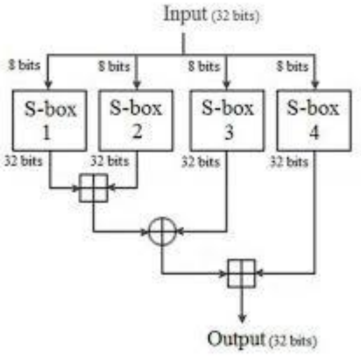


**Figure 2. Function F of Blowfish Algorithm**

John Kesley could only break 3-round of Blowfish, and his cryptanalysis cannot extend beyond 3-rounds. Serge Vaudenay examined a simplified variant of Blowfish, with the S-boxes known and not key-dependent. For this variant, a differential attack can recover the P-array with 2(8r+1) chosen plaintexts, where r is the number of rounds[9]. This attack is impractical in reality and does not work against 8-round Blowfish and higher. Since more plaintext is required than possibly be generated with a 64-bit block cipher[4], [5]. In 1996, Vincent Rijmen proposed a promising attack in his doctoral dissertation, but it can only break four rounds of Blowfish and no more[10]. Bruce Schneier shows blowfish possible differential cryptanalysis against reduced number-of-rounds or with the piece of information which describes the function F.

## 2.3 Fisher-Yates Shuffle

The Fisher-Yates shuffle (FYS), an algorithm for generating of the random permutation of a finite linear array and is suggested by Ronald Fisher and Frank Yates [19], [20]. Every permutation of FYS is unbiased in an array to produce equally alike. Richard Dursntenfeld introduces the modern version of Fisher-Yates shuffle Durstenfeld [18] and popularized by Donald E. Knuth in his pioneer book The Art of Computer Programming[19]. The modern version is an in-place shuffle, efficient requiring only time proportional to the number of elements shuffled and no additional storage space.

---To shuffle an array a of n elements (indices 0...n-1):

```
for i from n-1 down to 1 do
    j ← random integer such that $0 \leq j \leq i$
    exchange a[j] and a[i]
```

The basic method based from FYS, given for generating a random permutation of the numbers 1 through N. Procedure 1 showing the ways:

1. List a number from num 1 to N numbers.
2. Select a randomized number k between 1 and the list of remaining unshuffled numbers (included within).
3. Identify from the low-end list and mark out the kth number which is not shuffled out, and write down at the end of a separate list.
4. Go back to step 2 until all the numbers have been shuffled out.
5. The sequence of numbers written down in step 3 is now a random permutation of the original numbers.

**Procedure 1.** The basic method of FYS

Provides the random numbers pick in step 2 which are unbiased and truly random, so with the resulting permutation be. It suggests that the possibility of using a more straightforward method- selecting random numbers from one to N and discarding any duplicates- to generate the first half of the permutation and only applying the more complex algorithm to the remaining half.

## 3. CONTRIBUTION

Blowfish Algorithm has been exceptional in contrast to the other block cipher algorithms. In spite of its success, the S-Box permutation and F function need to enhance by improving its random permutation which significantly affected by the encryption process of the algorithm. Modifying Blowfish Algorithm is conceivable by using Fisher-Yates shuffle algorithm (FYS) to enhance the random generation and permutation value of S-box[21] and strengthening parallel execution of F function an additional key (user input key) added after performing the operators of S-boxes to further come up with a more robust and satisfying result. Figure 3 showing the modified structure of Function F.

### 3.1 A procedure of the S-box permutation

Table 2 showing the proposed substitution box using FYS algorithm

A.1. Initialize a linear array of SI of size 256 with values starting 0-256 in ascending order. Set $\Delta$ = length (SI)

A.2. Iterate the Piece-wise linear chaotic map for N0 times to die-out transient effect of the map with selected initial conditions.

A.3. Set counter= 1.

A.4. Further, iterate the map (1) and sample the chaotic state-variable x.

A.5. Extract a random number m $\in$ [1, k] from x as:

$m = \{floor(x*1010)\}mod(k) + 1$ where, $k= \Delta$ -counter+1

A.6. Exchange the two elements of array SI at positions m and k i.e $SI(k) \leftrightarrow SI(m)$.

A.7. Set counter = counter + 1, if counter < 256 go to A.4.

A.8. re-apply the FYS on current SI by repeating steps A.3 to A.7 $\xi$ times.

A.9. Translate resultant shuffled linear array SI to 16×16 table to get the final S-box.

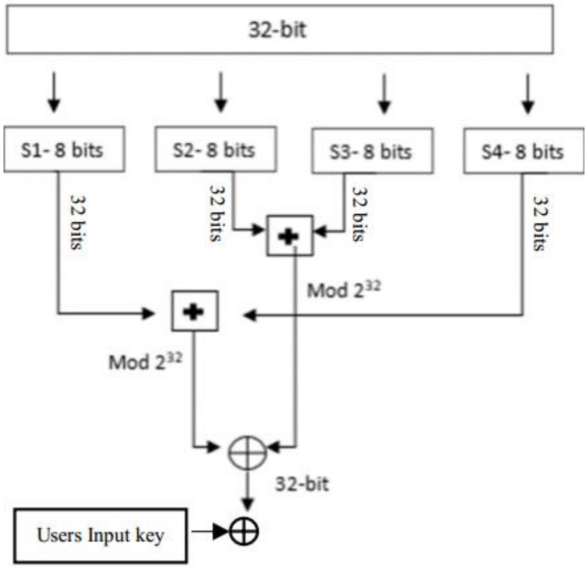


**Figure 3. Modified Function F of Blowfish Algorithm**

### 3.2 The Modified Function F

In equation 2 of Function F showing modification without violating the security requirements of the BF. The change supports the parallel evaluation of 2-OR operations ($S_{1,a}$ + $S_{4,d}$) mod $2^{32}$) and ($S_{2,b}$ + $S_{3,c}$) mod $2^{32}$) and with the use of 2-XORs: one for the parallel operation and another XOR for the users' input key. The users' input key further strengthens the S-box in the encryption process.

$$F(XL) = (((S_{1,\,a} + S_{4,d}) mod2^{32})\ XOR\ ((S_{2,b} + S_{3,c})\ mod2^{32})))XOR\ KI$$

**S**- Substitution box; **KI**- users Input key

**Equation 2:** Proposed Modified Function F

# 4. EXPERIMENTAL EVALUATION

The comparison and performance of the Modified Approach of Blowfish Algorithm with the Original Blowfish Algorithm. For evaluation, the following criteria are throughput, encryption time and decryption time. During the execution of both Algorithm was recorded using the different file formats like a word document, spreadsheet, picture and a portable document format (pdf).

Testing was done in the same platform using the C# programming language to obtain the fair comparison. A simulation was performed using a Laptop computer with an AMD E2 1.70 GHz processing speed, 4GB RAM, and 1TB internal memory with 80% free disk space and a 64-bit operating system.

The experimental runs were performed using the different file formats like word, spreadsheet, picture and pdf. Files have different contents.

## 4.1 Comparative Performance

The performance of the Modified BF and the Original BF was executed and operated for the different file formats and size to determine the processing speed. The result of the encryption and decryption and throughput are shown in figures 4,5,6,7 and 8 respectively.

From the figures, 4-7 it can be seen that Modified BF consistently has the least processing time among the files used regardless of its content and size.

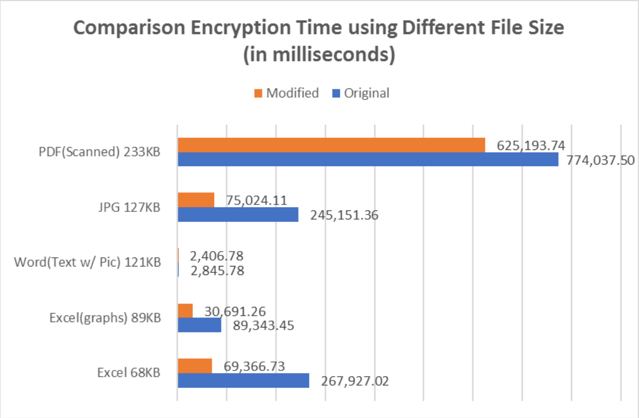


**Figure 4. Comparison of Encryption Time in Different File size (in milliseconds)**

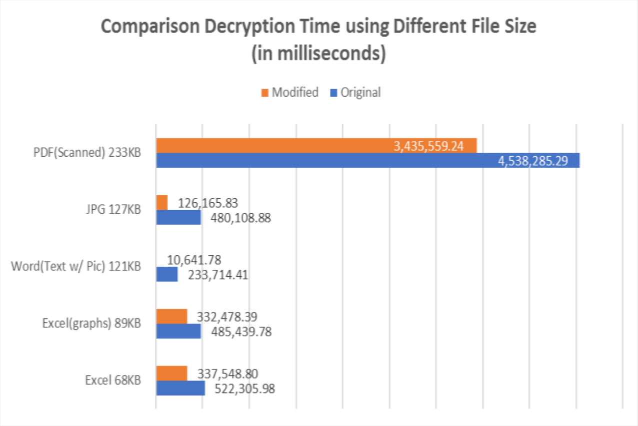


**Figure 5. Comparison of Decryption Time in Different File size (in milliseconds)**

In Figure 8 showing the efficiency of the Modified Blowfish Algorithm in encrypting any file type and size. The line graph shows that throughput for the Original Blowfish is lesser than the Modified Blowfish which means that the lower the throughput, the less efficient, the higher the performance, the higher the efficiency of encrypting any files and size.

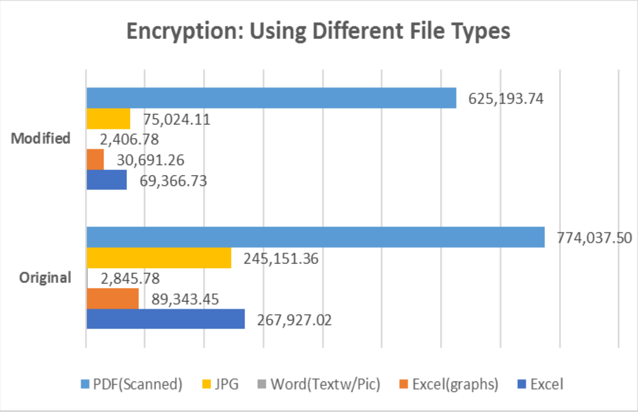


**Figure 6. Encryption: Showing the different File Types**

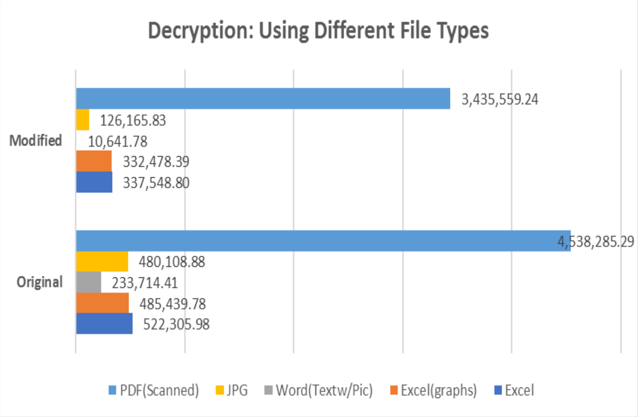


**Figure 7. Decryption: showing the different File Types**

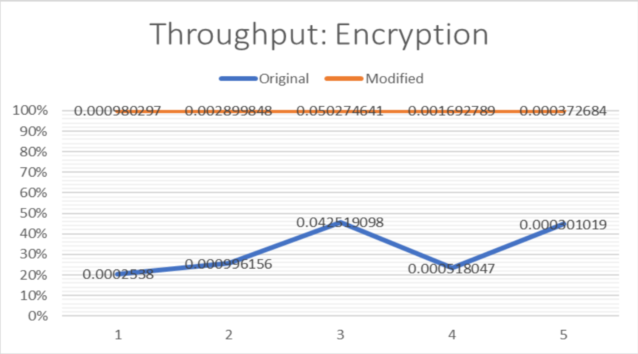
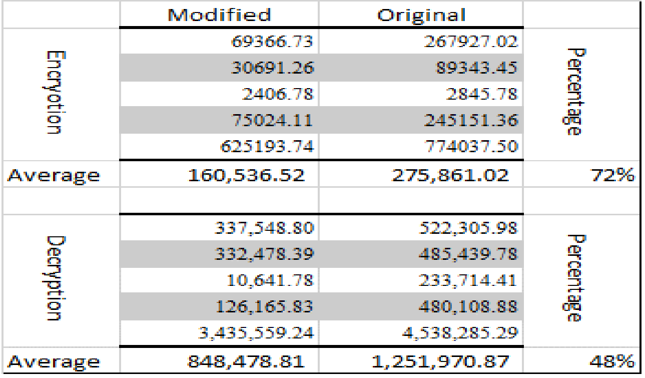


**Figure 8. Throughput**

## 4.2 Comparative Improvement

The use of YFS and modified Function F improved the processing time of the Blowfish Algorithm. The improvement over the BF using the existing Function F and Random Permutation in S-Box showing in Table 1. From the table, it clearly shows that the performance of Modified BF is incomparable to that of the Original BF, revealing a massive difference in Encryption time with a percentage of 72% with that of the processing of all the files used encrypted. So as with the Decryption time of 48% respectively.

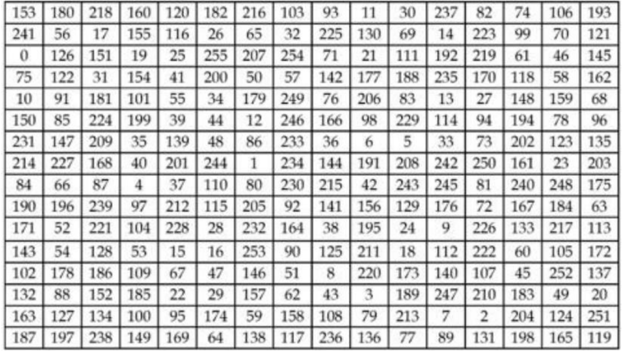
**Table 1. Comparative improvement of BF using YFS and Modified Function F regarding encryption and Decryption time (in milliseconds)**



| | Modified | Original | |
|---|---|---|---|
| Encryption | 69366.73 | 267927.02 | Percentage |
| | 30691.26 | 89343.45 | |
| | 2406.78 | 2845.78 | |
| | 75024.11 | 245151.36 | |
| | 625193.74 | 774037.50 | |
| Average | 160,536.52 | 275,861.02 | 72% |
| | | | |
| Decryption | 337,548.80 | 522,305.98 | Percentage |
| | 332,478.39 | 485,439.78 | |
| | 10,641.78 | 233,714.41 | |
| | 126,165.83 | 480,108.88 | |
| | 3,435,559.24 | 4,538,285.29 | |
| Average | 848,478.81 | 1,251,970.87 | 48% |

A sample output of the procedure permutation in 3.1, Table 2 displayed the permuted S-box using FYS Algorithm.

**Table 2. Proposed Substitution box**



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 153 | 180 | 218 | 160 | 120 | 182 | 216 | 103 | 93 | 11 | 30 | 237 | 82 | 74 | 106 | 193 |
| 241 | 56 | 17 | 155 | 116 | 26 | 65 | 32 | 225 | 130 | 69 | 14 | 223 | 99 | 70 | 121 |
| 0 | 126 | 151 | 19 | 25 | 255 | 207 | 254 | 71 | 21 | 111 | 192 | 219 | 61 | 46 | 145 |
| 75 | 122 | 31 | 154 | 41 | 200 | 50 | 57 | 142 | 177 | 188 | 235 | 170 | 118 | 58 | 162 |
| 10 | 91 | 181 | 101 | 55 | 34 | 179 | 249 | 76 | 206 | 83 | 13 | 27 | 148 | 159 | 68 |
| 150 | 85 | 224 | 199 | 39 | 44 | 12 | 246 | 166 | 98 | 229 | 114 | 94 | 194 | 78 | 96 |
| 231 | 147 | 209 | 35 | 139 | 48 | 86 | 233 | 36 | 6 | 5 | 33 | 73 | 202 | 123 | 135 |
| 214 | 227 | 168 | 40 | 201 | 244 | 1 | 234 | 144 | 191 | 208 | 242 | 250 | 161 | 23 | 203 |
| 84 | 66 | 87 | 4 | 37 | 110 | 80 | 230 | 215 | 42 | 243 | 245 | 81 | 240 | 248 | 175 |
| 190 | 196 | 239 | 97 | 212 | 115 | 205 | 92 | 141 | 156 | 129 | 176 | 72 | 167 | 184 | 63 |
| 171 | 52 | 221 | 104 | 228 | 28 | 232 | 164 | 38 | 195 | 24 | 9 | 226 | 133 | 217 | 113 |
| 143 | 54 | 128 | 53 | 15 | 16 | 253 | 90 | 125 | 211 | 18 | 112 | 222 | 60 | 105 | 172 |
| 102 | 178 | 186 | 109 | 67 | 47 | 146 | 51 | 8 | 220 | 173 | 140 | 107 | 45 | 252 | 137 |
| 132 | 88 | 152 | 185 | 22 | 29 | 157 | 62 | 43 | 3 | 189 | 247 | 210 | 183 | 49 | 20 |
| 163 | 127 | 134 | 100 | 95 | 174 | 59 | 158 | 108 | 79 | 213 | 7 | 2 | 204 | 124 | 251 |
| 187 | 197 | 238 | 149 | 169 | 64 | 138 | 117 | 236 | 136 | 77 | 89 | 131 | 198 | 165 | 119 |

## 5. CONCLUSION

The use of the YFS algorithm and modification of Function F has been able to improve the performance of the Blowfish Algorithm by reducing processing time.

Moreover, YFS and Function F modification was shown to perform consistently better compared to the Original Blowfish. It denotes that the shuffle algorithm is perfect, for it shows beyond expectation with a percentage of 72% of encryption time for the five files encrypted and 48% for decryption time respectively. Throughput also indicates the Modified BF a big gap to the Original BF. The study of future work is to implement in Cloud Computing for testing and evaluation.

## 6. FUTURE WORKS

This work is an avenue to test further the Modified BF and included in Cloud computing as a continuation of my study in my Doctorate Dissertation and to exploit to a real-world application.

## 7. REFERENCES

[1] M. Vanitha and R. Mangayarkarasi, "Comparative study of different cryptographic algorithms," *Int. J. Pharm. Technol.*, vol. 8, no. 4, pp. 26433–26438, 2016.

[2] B. Schneier, "Cryptographic design vulnerabilities," *Computer (Long. Beach. Calif).*, vol. 31, no. 9, pp. 29–33, 1998.

[3] S. Oukili and S. Bri, "High Throughput Parallel Implementation of Blowfish Algorithm," vol. 2092, no. 6, pp. 2087–2092, 2016.

[4] B. Schneier, "Description of a new Variable-Length Key, 64-Bit Block Cipher (Blowfish)," *Dr. Dobb's Journal, April 1994*, 1994.

[5] B. Schneier, "The {Blowfish} encryption algorithm," *Dr. Dobb's J. Softw. Tools*, 1994.

[6] B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," *Fast Softw. Encryption. Lect. Notes Comput. Sci. Cambridge Secur. Work. Proc. (December 1993)*, vol. 809, no. December 1993, pp. 191–204, 1994.

[7] A. Alabaichi, F. Ahmad, and R. Mahmod, "Security analysis of blowfish algorithm," *2013 2nd Int. Conf. Informatics Appl. ICIA 2013*, pp. 12–18, 2013.

[8] T. Nie and T. Zhang, "A study of DES and blowfish encryption algorithm," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 1–4, 2009.

[9] S. Vaudenay and E. N. S. Dmi, "On the Weak Keys o f Blowfish."

[10] V. Rijmen, "Cryptanalysis and design of iterated block ciphers.... - Google Scholar," 1997.

[11] R. Zhang and L. Chen, "A block cipher using key-dependent S-box and P-boxes," *IEEE Int. Symp. Ind. Electron.*, pp. 1463–1468, 2008.

[12] P. Mroczkowski, "Generating Pseudorandom S-Boxes – a Method of Improving the Security of Cryptosystems Based on Block Ciphers," *J. Telecommun. Inf. Technol.*, vol. nr 2, pp. 74–79, 2009.

[13] B. K. Maram and J. M. Gnanasekar, "Evaluation of Key Dependent S-Box Based Data Security Algorithm using Hamming Distance and Balanced Output," *Journal*, vol.

5, no. 1, pp. 67–75, 2016.

[14] Ü. Çavuşoğlu, A. Zengin, I. Pehlivan, and S. Kaçar, “A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system,” *Nonlinear Dyn.*, vol. 87, no. 2, pp. 1081–1094, 2017.

[15] D. Lambić, “A novel method of S-box design based on discrete chaotic map,” *Nonlinear Dyn.*, vol. 87, no. 4, pp. 2407–2413, 2017.

[16] K. Mohamed, M. Nazran, and M. Pauzi, “Study of S-box Properties in Block Cipher,” no. I4ct, pp. 362–366, 2014.

[17] D. Lambić and M. Živković, “Comparison of random S-Box generation methods,” *Publ. l’Institut Math.*, vol. 93, no. 107, pp. 109–115, 2013.

[18] M. Eberl, “The Fisher – Yates shuffle,” pp. 1–9, 2018.

[19] A. Olu, “A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures,” *Int. J. Comput. Appl.*, vol. 54, no. 11, pp. 975–8887, 2012.

[20] P. E. Black, “Fisher-Yates shuffle,” *Dictionary of Algorithms and Data Structures*, vol. 19. 2005.

[21] M. Ahmad, P. M. Khan, and M. Z. Ansari, “A Simple and Efficient Key-Dependent S-Box Design Using Fisher-Yates Shuffle Technique,” *Commun. Comput. Inf. Sci.*, vol. 420 CCIS, pp. 540–550, 2014.