Enhance the Performance and Security of Blowfish Cipher by modifying the algorithm

Ashokkumar Kothandan
X19138857
MSc in Cybersecurity

Abstract - The communication and transmission of data have its major part in this emerging world. The main concern of the communication medium is security to control the data breach, which is achieved by using a strong encryption algorithm. The objective of this paper is to enhance the performance of the Blowfish algorithm for fortifying security. Blowfish is a symmetric block cipher with a 64-bit data block cipher. Blowfish is the most recommended algorithm by satisfying the basic requirement in cryptography with its high insusceptibility and low algorithm complexity. This paper focuses on the modification of the F function, S Box and P Box functionality for the performance enhancement. Proposed approach results in the improvement in the performance of the algorithm in contrast to the traditional algorithm.

Keywords – Blowfish, Cryptography, Symmetric key, Block cipher.

Contents

1. INTRODUCTION

Cryptography is an art of securing the message transmitted in a channel. Modern cryptography is “the study of mathematical techniques for securing digital information, systems, and distributed computations against adversarial attacks” [1]. The usage of digital devices and digital communication has a tremendous increase in all the field including financial institution, marketing, etc. The potential growth of digital communication has increased security issues and exploits against the data. Thus, the development of cryptographic algorithm has high priority and challenging research. The key integral of the cryptography is confidentiality, integrity, and authentication [2].

A Block cipher is a type of cryptographic cipher where length ciphertext is the same length of the plain text [3]. Blowfish cipher is a kind of block cipher designed by Bruce Schneier in the year 1993 available as open source [4]. Though The blowfish has 64-bit length it was optimized for 32-bit CPUs. Blowfish algorithm has 16 rounds Feistel network where the 64 bits of plain text is converted to the same size of the ciphertext using an encryption key of length varying between 32 bit and 448 bits [7]. The 64-bit block is split into two halves with 32-bit each. Blowfish also uses 18 number of 32-bit P-array representing P1 to P18. Each P variable is XORed with the key and used in the F function [4]. The internal operation of F-Function is the input value is split into 4 pieces of 8-bits and all of them are sent to the S-Boxes separately to convert them into 32-bit and the final output is calculated with AND and XOR functions. S-Box is used to perform a byte by byte substitution of the. The output of the final ciphertext is calculated after the 16 rounds of iteration as two 32 bits and joined finally [5].

The main motivation of this research is to enhance the performance of the blowfish cipher in the aspects of speed and security. This can be achieved by modifying the function F or the S-Box. We will try to improve the security and reduce the number the rounds the iteration runs. The problem with the blowfish is the whole encryption process is influenced directly by S-Box.

To learn about the standard blowfish algorithm and its performance we will be researching the previous works on this topic. In the literature review we will be discussing the previous work on the cryptography and its importance, followed by the types of a brief description on the stream cipher and block cipher. Then we will be overviewing the blowfish algorithm and how it works. The complete mechanism is discussed to understand its concept and the performance values are measured. Later we will be discussing the limitations of the blowfish cipher and improving it to the current standard. We will also discuss the modification that can help in the enhancement of this algorithm.

2. LITERATURE REVIEW

To perform the enhancement of the blowfish cipher or any other research it is always important to start the learning from the basics thus in the literature review we will be beginning our research in terms of the cryptography and other important terms we need to know on completing this research successfully.

2.1 Modern Cryptography

As discussed earlier, cryptography is a study of the encryption and decryption technique where the data is transferred safely preventing the intrusion from unknown. Modern cryptography has been developed to overcome the security issues and the limitations in the classical cryptography [1]. Most of the classical cryptographic algorithm was based on the unproven assumption. The three important principles of the modern cryptography are as follows.

Principle 1- Formal definition: It should be impossible for the attackers to retrieve any kind of information from the ciphertext including the key or a part of the plain text [2].

Principle 2- Precise Assumption: Any assumption made be explicitly and mathematically precise. Validation of assumptions, comparison between two schemes, Understanding the necessity is preconceived [2].

Principle 3- Proof of Security: There are countless examples of unproven schemes that were broken, sometimes immediately and sometimes years after being developed [2].

According to Kerckhoff’s principle, “a cryptosystem should be secure even if everything about the system, except the key, is public knowledge”. Blowfish cipher is designed based on Kerckhoff’s principle as the algorithm is available public [1].

2.1.1 Symmetric Key vs Asymmetric Key

The symmetric key is an encryption scheme where the sender and the receiver should know the secret key to encrypt and decrypt any message transmitted [3]. Some of the classic examples of the symmetric key are DES, AES, etc including Blowfish. Blowfish uses the same key to encrypt and decrypt the message with the algorithm being exposed in public. The asymmetric key is a type of encryption scheme where a pair of keys are used to encrypt and decrypt the information. Regardless of whether the public key is known by everybody which is used to encrypt the message whereas only the receiver can decrypt it back to the original word using the private key.

2.1.2 Block Cipher

Block Cipher uses the symmetric technique where the size of the plain text is the same as the size of the ciphertext because it takes the plain text blocks at a time whereas the stream cipher takes 1 byte of plain text at a time [6]. Block cipher uses 64 bits at a time. The last block will be padded with redundant data such that the size of this block has 64 bits. For example, a final block of 38 bit will be padded with 26 more bit to make it a complete block for processing. Some of the examples of the block cipher are DES, AES, IDEA, Blowfish, etc [6].

2.2 Blowfish Algorithm

Blowfish algorithm is a 64-bit data symmetric block cipher with a variable-length private key that varies from 32 to 448 bits in range [4]. This protocol follows two fragments, one is an expansion of the key and the other is the encryption of data. The key expansion extends the 448 bits into several subnet key array of 4168 bytes. The data encryption is achieved with a 16 round of Feistel Network [4]. Every 64-bit block message is divided into two parts. Later we will be seeing the detail of explanation on how the two fragments work in this algorithm.

2.2.1 Key Expansion

The blowfish algorithm has a huge number of subnet keys. Total of 18 P-arrays used for each block for encryption and decryption. The exact process of the blowfish expansion of keys is as follows. with the help of the hexadecimal value of the pi for the initializing the P-array and the S-Boxes. The 64-bit message block is divided into two 32-bit halves where the first 32 bit is XORed with P1 and the next 32-bit is XORed with P2 [10]. The cycles continue, removing all the components of the P-array, and then all the S-Boxes one by one, with constantly shifting the output of the blowfish algorithm.

2.2.2 Encryption of Data

The blowfish algorithm is a Feistel network comprises of 16 circles. It has a 64-bit plain text input and an 18 P-array subnet key of 32-bit each as input. 64-bit ciphertext production.

INPUT: The message to be encrypted and P-array(P1 to P18)
Output: Encrypted message

The message or the plain text is divided into multiples of 64-bit blocks and the last block is padded with redundant data. Each 64-bit block is encrypted in the blowfish algorithm separately. The 64-bit block is divided into two halves with 32-bit each, LE_0 (first 32 bit) and RE_0 (second 32 bit) [10].

The blowfish algorithm as 16 rounds of iteration for each process will be the same as shown in figure 1.

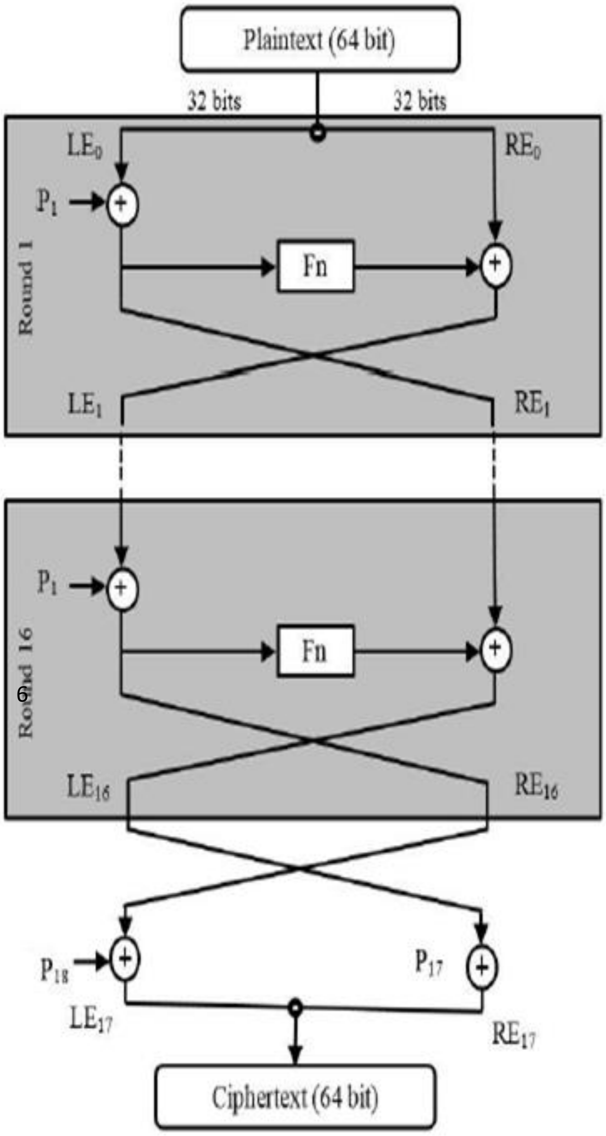


fig.1 Blowfish Algorithm

2.2.3 Algorithm

For each Round, the left 32-bit is XORed with the Pi from P-array where ‘i’ is the iteration number i.e., P1 for the 1st round P2 for the 2nd and so on. Followed by the output from the XOR is given as input in the function Fn and the output from the function Fn is XORed with the right 32-bit input [9].

The inputs of the next iteration will be the LE_i and RE_i where LE_i is the output from the second XOR and RE_i the output of the first XOR as shown in the figure. The iteration continues for a total of 16 times and finally the Last two outputs LE_{16} and RE_{16} are XORed with P17 and P18 and returns the final output RE_{17} and $LE_{17,}$ respectively[10]. The values are interchanged and merged to form the final ciphertext. This process is the same for all the blocks produced from the input plain text.

2.2.4 Function Fn

In every iteration, there is an Fn function that happens in the blowfish algorithm. In function Fn, the given 32-bit input from the XOR of LE_i and P_i is divided into 4 quarters of 8-bit input each is sent to the S-Box to make it as a 32-bit value[10]. S-Boxes or Substitution boxes in the blowfish algorithm are used to convert the 8-bit quarters into a 32-bit using substitution method as shown in figure 2. The output of the Function Fn is as below.

$$\text{Function Fn}= ((\mathbf{S_1} + \mathbf{S_2} \text{ MOD } \mathbf{2^{32}}) \text{ XOR } \mathbf{S_3}) + \mathbf{S_4} \text{ MOD } \mathbf{2^{32}}$$

Where S_1, S_2, S_3, S_4 are the outputs of 4 S-Boxes

S_1 and S_2 are performed AND function and MOD with 2^{32} to make it to 32-bit binary. The prior result is XORed with S_3 and XORed with S_4 with MOD 2^{32} produces the final output of the function Fn.

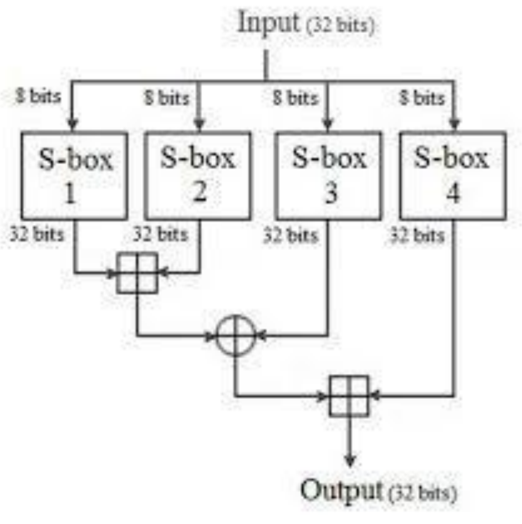


Fig 2: Function Fn with S-Box

Since the blowfish algorithm is a Feistel cipher, the decryption is the same as the encryption process with the P-array used in reverse order. Each half of the plain text is alternatively XORed with a round key and again XORed with the Function Fn. This is valid in decryption because the XOR is performed to the recent P_i.

2.2.5 Why to Modify blowfish algorithm

Though blowfish is the most recommended encryption technique, the key generation timeframe is more time consuming than the encryption of the message. This will be an advantage over the brute force attacks, but on this current situation 64-bit block size is not recommended as the size of the data transmitted is enormous than a decade [12].

The weak key is also one of the disadvantages of the blowfish algorithm where the key is generated from the Hexa-value of pi. The differential cryptanalysis attack is possible with two cases, either on reducing the number of rounds or a piece of a message describing the function Fn [12]. Comparing to the CAST cipher, the S-BOX are generated randomly in blowfish cipher.

3. RESEARCH METHODS

3.1 Proposed Architecture

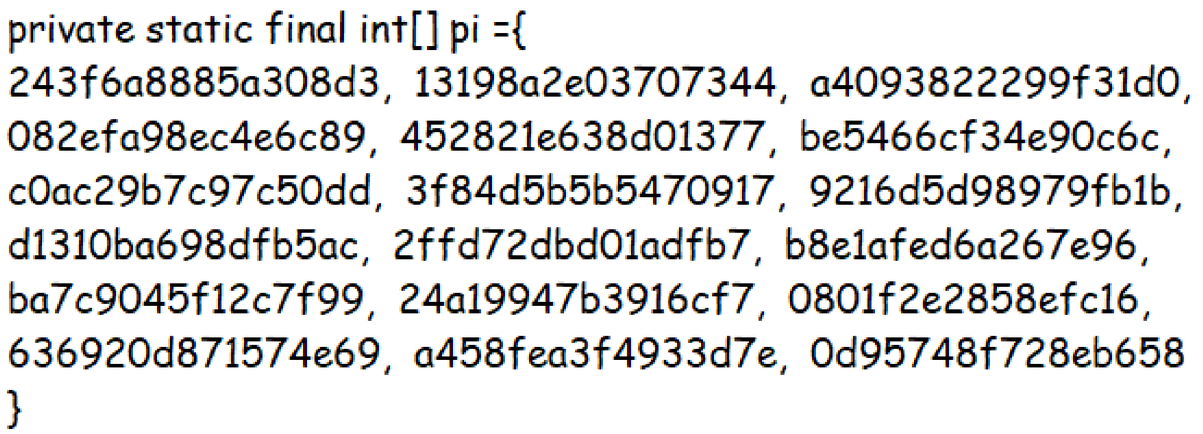
The proposal for the modification in the architecture of the blowfish algorithm based on the function Fn with efficiency in processing faster. The process consists of Initialization, Key generation, Modified function Fn. To improve the duration of the data encryption we will upgrade the block size from 64-bit to 128-bit.

3.1.1 Initialization

In this process, we will be initializing the data structures. The key matrix is a type of 4x4 two-dimensional matrix for the S-Box for the conversion of text from 32-bit to 64-bit. Another one-dimensional matrix of 18 elements for P-array each of 64-bits. We will be using the java platform for remodifying the algorithm.

3.1.2 Key Generation

P-array is a one-dimensional array of 64-bit size. This will be generated by a register in SSS with a small modification in the size of the register as the array size is increased from 32 to 64 bits. 18 elements are initialized with the value of pi i.e., 3.243f6a…. (hexadecimal value) [15].



```
private static final int[] pi ={
243f6a8885a308d3, 13198a2e03707344, a4093822299f31d0,
082efa98ec4e6c89, 452821e638d01377, be5466cf34e90c6c,
c0ac29b7c97c50dd, 3f84d5b5b5470917, 9216d5d98979fb1b,
d1310ba698dfb5ac, 2ffd72dbd01adfb7, b8e1afed6a267e96,
ba7c9045f12c7f99, 24a19947b3916cf7, 0801f2e2858efc16,
636920d871574e69, a458fea3f4933d7e, 0d95748f728eb658
}
```

Fig 3: Value of P-array from pi

3.1.3 Modified Function Fn

The process of the generation of the S-Box should differ from the normal method as the number of bit size is increased from 32 to 64-bit instead of 8 to 32-bit. This is because the function of blowfish is modified for performance enhancement. Here we will be using the 4 S-Boxes to encrypt the data. The function definition is

$$Fn_1 = S_1(I_L) \text{ XOR } S_2(I_R)$$

$$Fn_2 = S_3(Fn_{1L}) \text{ XOR } S_4(Fn_{1R})$$

Where Fn_2 is the final output of the function

S_1 ,S_2, S_3, S_4 are the 4 S-Boxes

I_L and I_R are the two halves of the 64-bit input in Fn

Fn_{1L} and Fn_{1R} are the two halves of the Fn_1

For the clear understanding of the Modified Fn refer to figure 3 as shown below

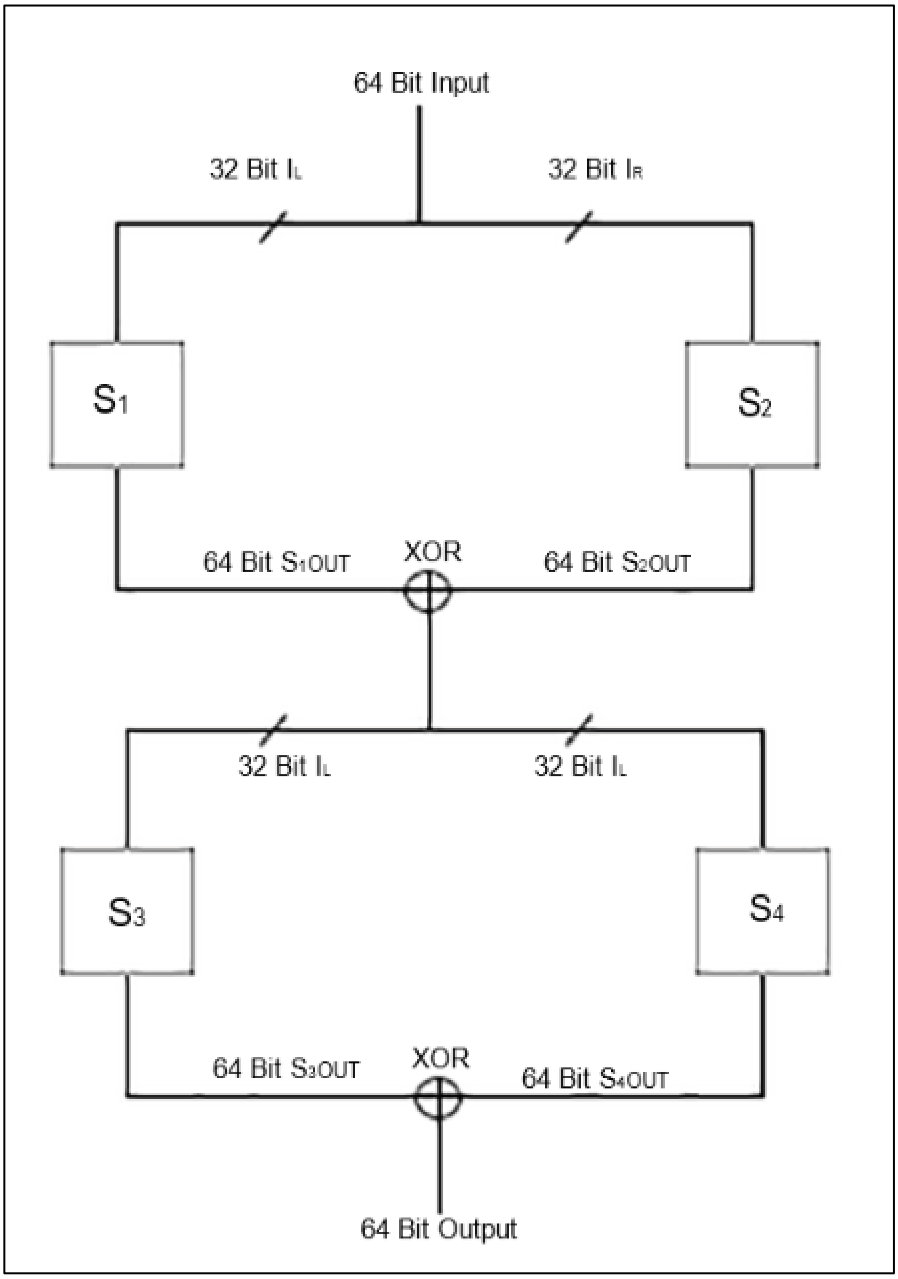


Figure 4: Modified Fn

3.2 Performance Analysis

To determine the performance of the modified blowfish algorithm and comparing it with the original blowfish algorithm, we will encrypt a grayscale bit image of 1kb size and analyse based on parameters like encryption speed, file size after encryption, correlation coefficient analysis [13].

3.2.1 Encryption Speed

One of the main performance analyses is based on the speed of encryption and decryption. A comparison test is run on both the original and proposed algorithm and time taken for each round is captured and analysed on them.

3.2.2 File Size Comparison

The size of the file encryption is compared on both original and proposed blowfish algorithm for different file types like jpeg, bmp, pdf, word, and excel. The efficiency of the modified algorithm is calculated based on lower throughput, higher efficiency after encryption.

3.2.3 Correlation coefficient Analysis

The correlation coefficient between two adjoining pixels of an image is calculated by using the formula

$$r = \frac{\mathrm{cov}(x,y)}{(D(x))^{0.5} * (D(y))^{0.5}}$$

where x and y are the value of adjoining pixels of the image

D(x) and D(y) are the differences of x and y values.

cov(x,y) is the covariance of x and y

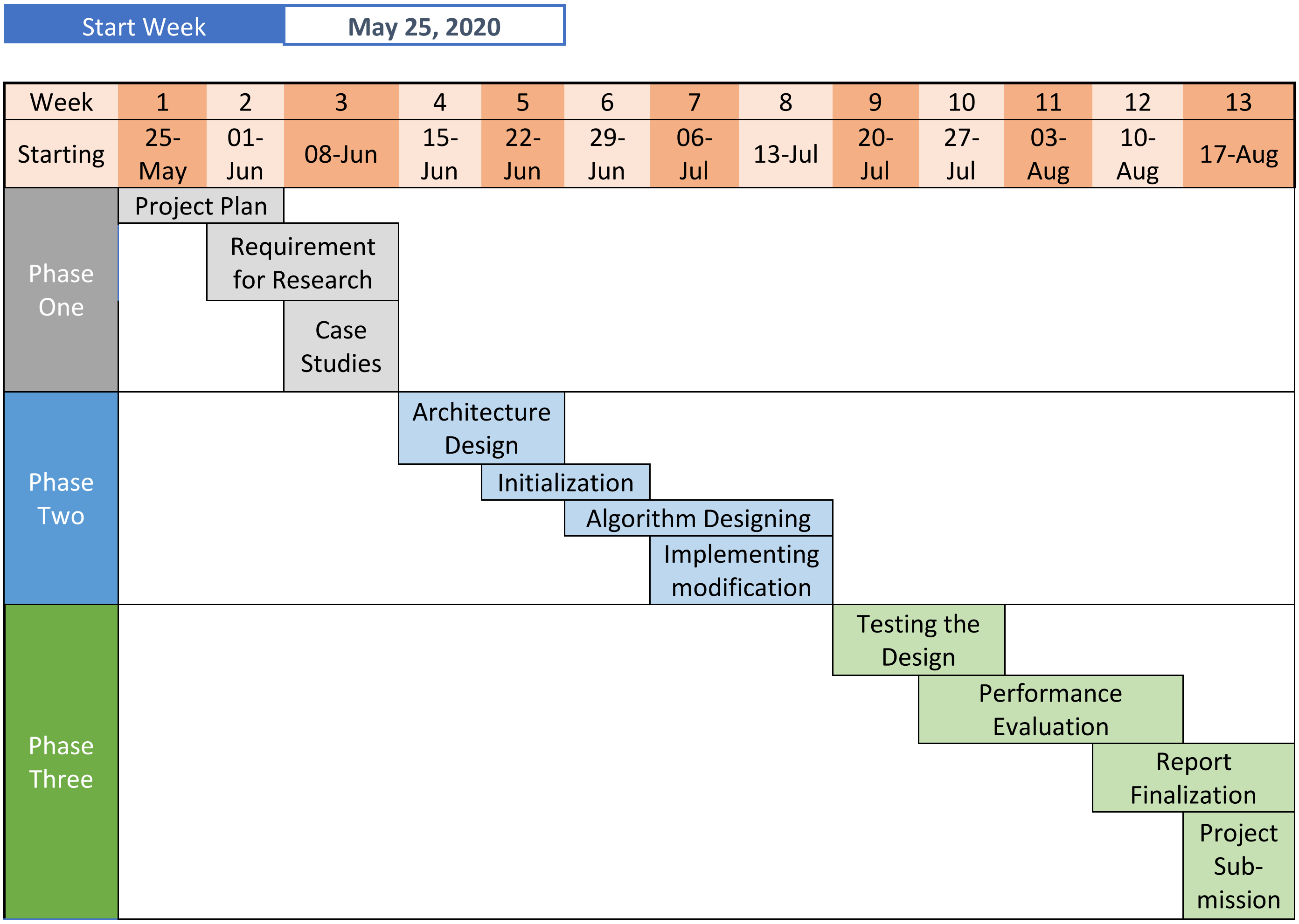
To determine the correlation coefficient, we select randomly 1000 pixels of the original image and adjoining pixel of the encrypted image. From the selected pixel we calculate the correlation coefficient for both the original and modified algorithm [15].

3.2.4 Key Sensitivity Test

The key test is based on encrypting the same message or data with different keys and identify the performance ratio of the keys for both the original and modified cipher, to determine the quality of the encryption on the key change. This test is mostly used to determine the time consumed on the key management feature [9].

4. Gantt Chart

Figure 5 represents the Gantt chart of the task lifecycle for the completion of the project work for the next semester.



Start Week	May 25, 2020

Week	1	2	3	4	5	6	7	8	9	10	11	12	13
Starting	25-May	01-Jun	08-Jun	15-Jun	22-Jun	29-Jun	06-Jul	13-Jul	20-Jul	27-Jul	03-Aug	10-Aug	17-Aug

Figure 5, Gantt Chart for Project Plan

5. Conclusion

The main objective of this research is to modify the blowfish cipher to enhance the performance and security of the cipher and analyse the performance by comparing the value based on different parameters. This research helps in learning more about the blowfish cipher and its algorithm and finding the weakness in them to improve it. Though we can’t conclude that the proposed methodology will improve the performance so future work may lead to some changes based on the improvement need for the algorithm to be successful.

6. Reference

[1] J. Katz and Y. Lindell, Introduction to modern cryptography. Boca Raton, FL: CRC Press, 2015.

[2] William Stallings, ”Cryptography and Network Security”, Pearson Publication, Prentice Hall, 2013.

[3] Y. Nawaz and L. Wang, "Block Cipher in the Ideal Cipher Model: A Dedicated Permutation Modelled as a Black-Box Public Random Permutation", Symmetry, vol. 11, no. 12, p. 1485, 2019.

[4] "Schneier on Security: The Blowfish Encryption Algorithm", Schneier.com, 2020. [Online]. Available: https://www.schneier.com/academic/blowfish/.

[5] "A Novel Encryption Algorithm by Fusion of Modified Blowfish Algorithm and Fermat’s Little Theorem for Data Security", International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 4, pp. 1188-1192, 2020.

[6] V. Agrawal and M. Darji, "Performance Comparison and Enhancement of Blowfish Algorithm", INTERNATIONAL JOURNAL OF RESEARCH IN ADVANCE ENGINEERING, vol. 1, no. 2, p. 12, 2017. Available: 10.26472/ijrae.v1i2.5.

[7] K. Kanagalakshm and M. Mekala, "Enhanced Blowfish Algorithm for Image Encryption and Decryption with Supplementary Key", International Journal of Computer Applications, vol. 146, no. 5, pp. 41-52, 2016.

[8] Ashwak ALabaichi and Faudziah Ahmad, "Security Analysis of Blowfish algorithm".

[9] Rekha C, Krishnamurthy G N, “An Optimized Encryption Algorithm and F function with dynamic substitution for creating S Box and P Box entries for blowfish algorithm”, international journal of scientific & technology research volume 8, issue 12, 2019.

[10] Vaibhav Poonia and Dr. Narendra Singh Yadav, “Analysis of modified Blowfish Algorithm in different cases with various parameters”, 2015 International Conference on Advanced Computing and Communication Systems (ICACCS -2015), Jan. 05-07,2015, 2015.

[11] M. Valmik and P. Kshirsagar, "Blowfish Algorithm", IOSR Journal of Computer Engineering, vol. 16, no. 2, pp. 80-83, 2014.

[12] Serge Vaudenay, “On the Weak Keys of Blowfish”.

[13] V. Agrawal and M. Darji, "Performance Comparison and Enhancement of Blowfish Algorithm", International Journal of Research In Advance Engineering, vol. 1, no. 2, p. 12, 2017.

[14] Jeyamala Chandrasekaran, B. Subramanyan, and Raman Selvanayagam, " A Chaos Based Approach for Improving Non-Linearity in S Box Design of Symmetric Key Cryptosystems".

[15] R. R. Corpuz, B. D. Gerardo, and R. P. Medina, “A Modified Approach of Blowfish Algorithm Based On S-Box Permutation using Shuffle Algorithm,” Proceedings of the 2018 VII International Conference on Network, Communication and Computing - ICNCC 2018, 2018.

[16] H. Poston and K. Dhandhania, "Blowfish: The first well-known encryption algorithm in public domain | CommonLounge", Commonlounge.com, 2020. [Online]. Available: https://www.commonlounge.com/discussion/d95616beecc148daaa23f35178691c35. [Accessed: 06- Apr- 2020].

[17] 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.