ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 2, Issue 7, July 2014**

# Optimized Blowfish Encryption Technique

Christina L [1], Joe Irudayaraj V S [2]

M. Phil Scholar, Department of Computer Science, St. Joseph College (Autonomous), Tiruchirapalli, Tamilnadu, India[1]

Associate Professor, Department of Computer Science, St. Joseph College (Autonomous), Tiruchirapalli, Tamilnadu, India[2]

**ABSTRACT**: The internet plays an important role in day-to-day life. The people can transfer important data through the internet such as Email, banking transaction and online purchase. In order to get secured transaction, network security is essential. Network security is mostly achieved through the use of cryptography. Cryptography refers to the art and science of transforming the message to make them secure and immune to attacks. Different algorithms and protocols are used to protect the data. The efficiency of the algorithm is measured by execution time and throughput. Using of larger key size may affect the efficiency of the algorithm. Blowfish is a symmetric block cipher with a 64 bit block size and variable key length from 32 bits to 448 bits. The Blowfish algorithm keeps two sub key arrays: four S-boxes and single P-box. Focus of this research is to optimize the four S-boxes into two S-boxes in original Blowfish algorithm to increase the speed and examine the effectiveness and limitations of some Block cipher algorithms. The program simulation result provides the better performance as well as security.

**KEYWORDS:** Blowfish, Optimized Blowfish, Encryption Time, Decryption Time, Throughput.

## I.INTRODUCTION

Cryptography is the science that is widely used for network security [2]. Cryptography means to transfer sensitive information across insecure networks such as internet [3]-[5]. Key aspects of cryptography are confidentiality, integrity, authentication, and non repudiation [1] [2]. An original message is known as the plaintext, while the coded message is called the ciphertext. The process of converting from plaintext to ciphertext is called encryption; restoring the plaintext from the ciphertext is decryption [6].

Cryptographic algorithm is classified into two categories: (i) Symmetric Key Cryptography where one key is used for both encryption and decryption. (ii) Asymmetric Key Cryptography where two different keys are used one for encryption and other for decryption [7]. Symmetric key cryptography is divided into two types on the basis of their operations [8]: (i) Stream Cipher: A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. (ii) Block Cipher: A block cipher is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length [6].

Blowfish is a symmetric block cipher designed by Brute Schneier in December 1993. Blowfish is a replacement of DES or IDEA [14]. Blowfish algorithm is a symmetric block cipher with a 64-bit block size and variable key length from 42 bits to 448 bits [9] [10].

## II.RELATED WORK

Manikandan G, Rajendran P, Chakarapani K, Krishnan G and Sundarganesh G (2012) modified the Blowfish algorithm for enhancing data security. They designed a software tool for encrypting the file. A file is divided into a number of pieces depends upon the users specification and then the encryption algorithm is applied. To enhance the performance of the software they modified the F-function in Blowfish algorithm. The F-function consists of 4 S- boxes (S1, S2, S3 and S4). The F-function is defined as F(X)=(( S1+ S2 mod 232) XOR S3+ S4 mod 232. They modified this F-function into F(X) = ((S1 XOR S2 mod 232) + (S3 XOR S4 mod 232)). They experimentally proved that the execution time of modified Blowfish is 14% lesser than the original Blowfish algorithm [8].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 2, Issue 7, July 2014**

Monika Agrawal and Pradeep Mishra (2012) enhanced the security level of Blowfish algorithm and decrease the time for encryption and decryption. They generate the random number within the range 0 to 65535. Set the flag value to zero. Convert random number into 16 bit binary form and finds the positions that are holding 0 entries, change the flag value to one otherwise flag is zero. If the flag is 1 then the F-function will not work, it the flag is 0 then the F-function will work. In every round a new random number is generated and this as a result gives difference in the application of F function. They analysed that the encryption time and decryption time is reduced compare than original Blowfish algorithm [10].

Israa Tahseen and Shatha Habeeb (2012) proposed a new approach to generate a random number using image. Read the picture by pixel, select the specific location randomly pick any two colors. Apply XOR between two selected colors then specify the length of the key. This key is used to encrypt or decrypt the plaintext. They apply this key generation method in to Blowfish algorithm. Finally they suggest that this type of key generation is suitable for short keys in symmetric system [11].

Ratinder Kaur and V. K. Banga (2012) proposed a security of image using Blowfish algorithm. They divide the original image into a random number of blocks and then the blocks are transformed into new locations. For better transformation divide the blocks into a smaller number of blocks because fewer pixels keep their neighbours and difficult to predict neighbours pixels. For better performance divide the block into larger number then the correlation was reduced even further and the entropy was increased. After that transformation encrypt the block of images using Blowfish algorithm. Finally they analysed that combination of transformation and encryption provides better security for image [12].

B.Geethavani, E.V.Prasad and R.Roopa (2013) proposed a new approach for secured the data, transfer in audio signals using discrete wavelet transform. They derived to new hybrid technique from the combination of cryptography and stegnography for transmitting the message in a highly secured manner. The plaintext is encrypted using Blowfish algorithm and the resulted cipher text is embedded into an audio file using discrete wavelet transform. Finally they suggest this method is efficient method for hiding text in audio files such that data can reach the destination in a safe manner without any modification [13].

This paper is organized as follows: section III describes optimized Blowfish encryption algorithm and section IV provides the Pseudo code of Optimized Blowfish comparative analysis of original Blowfish and optimized Blowfish encryption algorithm based on execution time and throughput, and section V presents the conclusion and future enhancement.

## III. OPTIMIZED BLOWFISH ENCRYPTION ALGORITHM

Optimized Blowfish is a 64 bit block cipher with a 448 bit key length. It is a Feistel network consisting of 16 rounds. The relative strength of the encryption algorithm is based on key length. Optimized Blowfish algorithm keeps two sub key arrays: P-array and two 32 bit S-boxes. This algorithm is divided into three main parts: sub keys generation, S-Box preparation and Encryption.

Description of optimized Blowfish Encryption Algorithm

**Sub key generation (P-array):** Initialized the P-array with fixed string. It contains eighteen 32 bit sub key values. Separate the key string into eighteen 32 bit. The first P-array ($P_1$) value is XORed with the first 32 bit key ($K_1$), second P-array ($P_2$) value is XORed with the second 32 bit key ($K_2$), third P-array ($P_3$) value is XORed with the third 32 bit key ($K_3$) up to 18 rounds. That means each eighteen 32 bit P-array value is XORed with eighteen 32 bit key values. All zero string is encrypted using optimized Blowfish encryption algorithm. This process is executed in 18 rounds. After that the sub key values are stored in P-array.

**S-Box preparation:** Initialize the four S-boxes with fixed string. Each S-box consists of 256 entries. These S-box values are encrypted using Blowfish encryption algorithm. After that first and second S-box values are concatenated and third and fourth S-box values are combines together. Finally the four S-box values are reduced into two S-boxes.

**Data Encryption**: Data encryption consists of F function with 16 rounds. Each round consists of a key dependent permutation and a key-and-data dependent substitution. Each round every left half is affected by the right half as well

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

as every subkey is affected by every key. The figure 1 shows the structure of Blowfish algorithm. This structure is same as optimized Blowfish algorithm.

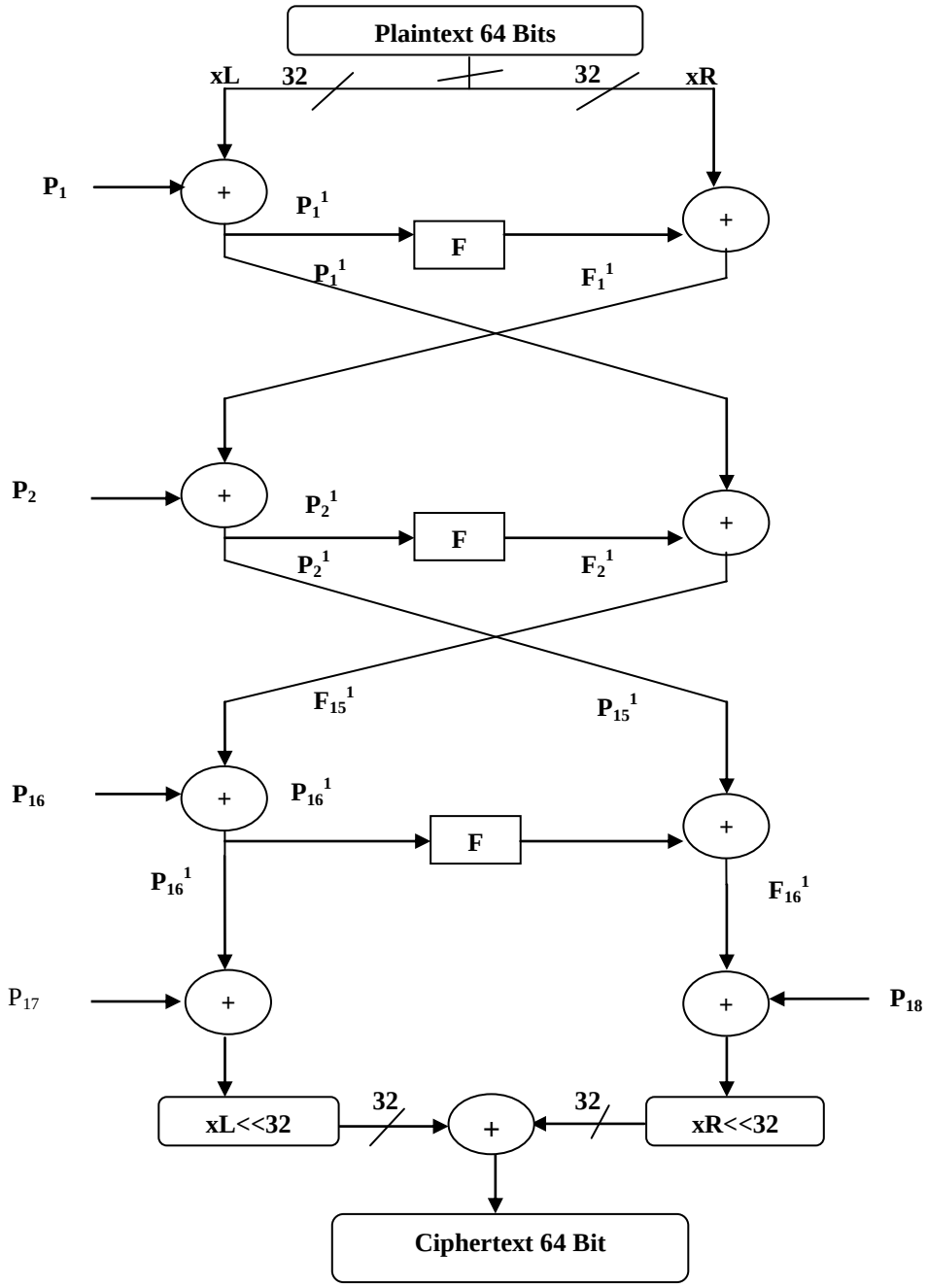



**Fig. 1 Blowfish Encryption**

Function F is not reversible, gives the best possible avalanche effect for a Feistel network. The original F-function consists of four S-boxes. The input of 32 bit is divided into four 8 bits. These four eight bits values are mixed using addition modulo and combined using XOR. The figure 2 illustrates the function F.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

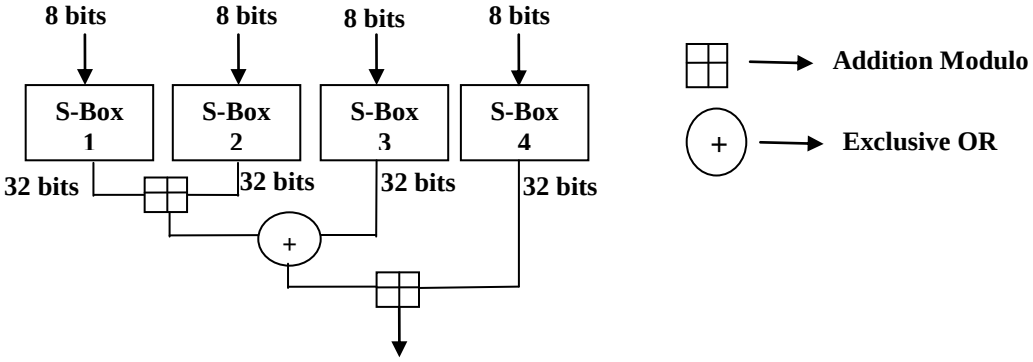



**Fig 2 Function F**

**Modified F function**

The only change is S-boxes in the F-function. The Feistel structure of Blowfish algorithm is not changed but the structure of F-function is modified. The original Blowfish algorithm F-function has four S-boxes but the optimized Blowfish F-function has two S-boxes. The figure 3 illustrates the modified structure of F-function.

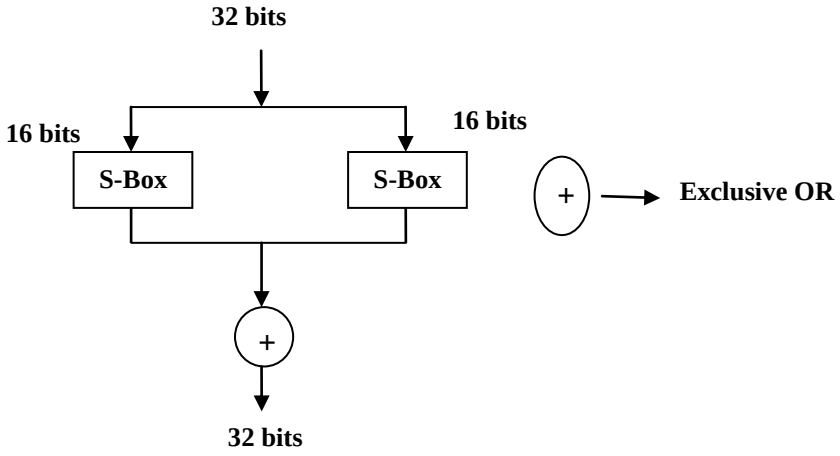



**Fig. 3 Optimized F- function**

The working of Optimized Blowfish Illustrated as follows

1. Initialized the P-array and four S-boxes, in order with a fixed string.
2. Encrypt the key and P-array for prepare the sub keys.
3. Encrypt the S-box values using F function with four S-boxes
4. Divide the 64 bit input data into two 32-bit halves (left and right). The left half is denoted by XL and right half is denoted by XR.
5. The 32 bit left half XL is XORed with the sub key $P_1$ and assigned into the XL. The XL is fed into the F function.
6. The F function consists of two S-boxes. F function splits the 32 bit of input into two 16 bit halves and each half is input to each S-boxes.
   - 6.1 The first 16 bit S-box ($S_1$) and second 16 bit S-box ($S_2$) are added.
   - 6.2 The 32 bit resulted bit is XORed.
   - 6.3 The optimized F function is as follows: Divide XR into two 16 bit halves: a, and b.

   F(XL) = F(a, b) = ($S_1$ ð $S_2$) Here "ð" is XOR.
7. F(XL) is XORed with XR

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 2, Issue 7, July 2014**

8. Interchange (Swap) the XL and XR values that means right half (XR) becomes new left half and left half (XL) become new right half (XR).
9. After the 17th round left half and right half are not swapped but the XR is XORed with $P_{17}$ and XL is XORed with $P_{18}$.
10. Finally XL and XR are recombined using exclusive OR.
11.

Decryption is same as encryption, but $P_0$, $P_1$,..., $P_{17}$ are used in the reverse order. In case of original F-function which executes in sequential order and it requires two Addition operations and one XOR operations. But in the case of optimized F-function requires only one XOR operation. To reduce four S-boxes entries in to two S-boxes cannot affect the security.

## IV. PSEUDO CODE

A. Pseudo-code of F-Function with four S-Boxes (S0, S1, S2 and S3)
Step 1: Divide xL into four eight-bit quarters: a, b, c, and d
Step 2: F(xL)=((S0,a+S1,b mod $2^{32}$)^S2,c)+S3,d mod $2^{32}$.
B. The Pseudo-code of optimized F function with two S-boxes
Step 1: Divide xL into two sixteen-bit quarters: a, and b.
Step 2: F(xR)=(S0,a^ S1,b)
C**.** Pseudo-code of Encryption
Step 1: Divide the 64 bit input data into two 32-bit halves (left and right): xL and xR
Step 2: for i=0 to16
xL is XORed with P[i].
Find F(xL)
F(xL) is XORed with xR.
Interchange xL and xR.
Step 3: Interchange xL and xR.
Step 4 : xR is XORed with P[16].
Step 5: xL is XORed with P[17].
Step 6: Finally combine xL and xR.
D. Pseudo-code of Decryption
Step 1: Divide the 64 bit input data into two 32-bit halves (left and right): xL and xR
Step 2: for i=17 to1
xL is XORed with P[i].
Find F(xL)
F(xL) is XORed with xR.
Interchange xL and xR.
Step 3: Interchange xL and xR.
Step 4: xR is XORed with P[1].
Step 5: xL is XORed with P[0].
Step 6: Finally combine xL and xR.

## V.SIMULATION RESULTS

A comparative analysis of original Blowfish and Optimized Blowfish is performed to provide some measurements on the encryption and decryption. The following parameters are used as the performance criteria, such as input size (in bytes), Encryption time, Decryption time and throughput.

These algorithms are implemented in Java jdk1.7 and developed using in Eclipse (4.2.0) integrated environment. Performance was measured on a Intel(R) Core(TM)2 DUO CPU T6600 @ 2.20 GHz processor with 3GB RAM running Windows 7 Home Basic 2009, Service Pack 1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 2, Issue 7, July 2014**

C. Performance Comparison on the basis of Execution time and Throughput

The comparison is conducted for the text or message only. Encryption time is one of the performance parameter, which is defined as the amount of time required for converting plaintext into ciphertext at the time of encryption. Decryption time is also one of the performance parameter, which is defined as the amount of time required for converting ciphertext into plaintext at the time of decryption. To improve the accuracy of the timing measurement, the program is executed in 10 times. The encryption time and decryption time is measured by milliseconds. The sum of encryption time and decryption time is considered as the execution time.

Execution time=Encryption time + Decryption Time. Table I shows the total execution time.

**TABLE I**
**Total Execution Time**

| Plaintext Size (Bytes) | Key Size (Bytes) | Original Blowfish | | Optimized Blowfish | | Original Blowfish | Optimized Blowfish |
|---|---|---|---|---|---|---|---|
| | | Encryption Time | Decryption Time | Encryption Time | Decryption Time | | |
| 41 | 8 | 5.42 | 0.82 | 5.00 | 0.89 | 6.24 | 5.90 |
| 82 | 12 | 6.4 | 1.51 | 6.19 | 1.45 | 7.91 | 7.64 |
| 47 | 16 | 5.42 | 0.87 | 5.15 | 0.86 | 6.29 | 6.01 |
| 70 | 20 | 6.01 | 1.31 | 5.89 | 1.37 | 7.32 | 7.26 |
| 45 | 24 | 5.24 | 0.87 | 5.27 | 0.85 | 6.11 | 6.12 |
| 72 | 28 | 6.30 | 1.26 | 6.23 | 1.31 | 7.56 | 7.54 |
| 57 | 36 | 5.94 | 1.19 | 5.77 | 1.09 | 7.13 | 6.86 |
| 52 | 40 | 5.92 | 1.07 | 5.65 | 0.99 | 6.99 | 6.64 |
| 49 | 44 | 5.91 | 1.07 | 5.74 | 1.03 | 6.98 | 6.77 |
| 63 | 56 | 6.53 | 1.20 | 6.36 | 1.17 | 7.73 | 7.53 |
| Average Execution Time | | | | | | 6.83 | 7.03 |

From the result identified that the average execution time of optimized Blowfish is 6.83 milliseconds and the original Blowfish is 7.03 milliseconds; comparison of these two average execution time, optimized Blowfish algorithm is less than original Blowfish algorithm.

D. Performance comparison on the basis of Throughput

The throughput of the execution scheme is calculated as the total encrypted plaintext size in bytes divided by the total execution time. Throughput indicates the speed of encryption. The throughput of the execution scheme is calculated using the following formula.

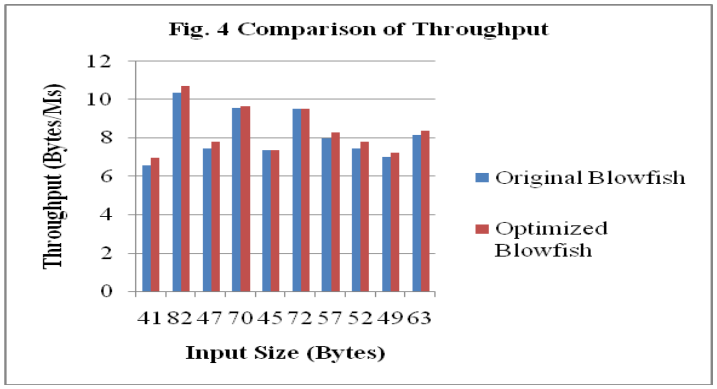
Throughput=Total size of plaintext /Total execution time. Where plaintext size is measuring in bytes and total execution time is measuring in milliseconds. Comparison of throughput of these algorithms is shown in Figure 4.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 2, Issue 7, July 2014**




The throughput of the execution scheme is calculated as the total encrypted plaintext size in bytes divided by the total execution time. Throughput indicates the speed of encryption. The above graph shows the result based on the throughput of the execution time with different input size. It shows that the throughput is high for optimized Blowfish when compared to original Blowfish algorithms. As the throughput value is increased, the execution time of encryption is decreased.

## VI.CONCLUSION

In this paper an optimized Blowfish has been developed. The longer key size is more secure but the encryption time and decryption speed is slow. In order to overcome this problem in Blowfish algorithm reducing of two S-boxes will increase the speed and provide the better security to data. The main advantage of optimized Blowfish is that the execution time is reduced to 0.2 milliseconds and the throughput is increased to 0.24bytes/milliseconds compare than original algorithms. In future, cryptanalysis of optimized Blowfish algorithm will be investigated and this algorithm is tested with other data type such as text file, audio and video.

## REFERENCES

1. Behrouz A. Forouzan, “Cryptography and Network Security”, Tata McGraw-Hill, 2nd edition, 2008.
2. Anand Kumar M and Dr. S. Karthikeyan, "Investigating the Efficiency of Blowfish and Rejindael Algorithms", International Journal of Computer Network and Information Security, pp. 22-28, 2012.
3. Obaida Mohammad Awad Al-Hazaimeh, "Design of a New Block Cipher Algorithm", Network and Complex Systems, Vol. 3, No. 8, pp. 1-5, 2013.
4. Md Imran Alam, "A Comparative Analysis of Different Encryption Techniques of Cryptography", International Journal of Advanced and Innovative Research, Vol. 2, Issue 9, pp. 160-166, 2013.
5. Ali M Alshahrani, "Different Data Block Size Using to Evaluate the Performance Between Different Symmetric Key Algorithms", International Journal of Computer Networks and Communications, Vol. 6, No. 2, pp. 89-97, 2014.
6. William Stallings, "Cryptography and Network Security", Fifth Edition, Pearson Publication, Prentice hall, 2013.
7. Neeraj Khanna et al, "New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm, IEEE, pp.125-130, 2011.
8. Manikandan G, Rajendran P, Chakarapani K, Krishnan G and Sundarganesh G, "A Modified Crypto Scheme for Enhancing Data Security", Journal of Theoretical and Applied Information Technology, Vol. 35, No.2, pp.149-154, 2012.
9. Ashwak Alabaichi, Faudziah ahmed and Ramlan Mahmod, "Security Analysis of Blowfish algorithm", IEEE, pp. 12-18, 2013.
10. Monika Agrawal and Pradeep Mishra, "A Modified Approach for Symmetric Key Cryptography Based on Blowfish Algorithm", International Journal of Engineering and Advanced Technology, Vol. 1, Issue 6, pp. 79-83, 2012.
11. Israa Tahseen and Shatha Habeeb, "Proposal New Approach for Blowfish Algorithm by Using Random Key Generator", Journal of Madent Alelem College,Vol. 4, No. 1, pp. 1-10, 2012.
12. Ratinder Kaur and V. K. Banga,"Image Security using Encryption based Algorithm", International Conference on Trends in Electrical, Electronics and Power Engineering (ICTEEP'2012), pp. 110-112, 2012.
13. B.Geethavani, E.V.Prasad and R.Roopa, "A New Approach for Secure Data Transfer in Audio Signals Using DWT", IEEE, 2013.
14. Bruce Schneier, “Applied Cryptography”, John Wiley & Sons, 2nd Edition, New York, 1996.