# A Modified Approach for Symmetric Key Cryptography Based on Blowfish Algorithm

**Monika Agrawal, Pradeep Mishra**

***Abstract— The principal goal of designing any encryption algorithm is to hide the original message and send the non readable text message to the receiver so that secret message communication can take place over the web. The strength of an encryption algorithm depends on the difficulty of cracking the original message. A number of symmetric key encryption algorithms like DES, TRIPLE DES, AES, BLOWFISH has been developed to provide greater security affects one over the other. Although the existing algorithms have their own merits and demerits but this paper presents a new approach for data encryption based on Blowfish algorithm. The blowfish algorithm is safe against unauthorized attack and runs faster than the popular existing algorithms. With this new approach we are implementing a technique to enhance the security level of blowfish algorithm and to further reduce the time for encryption and decryption.***

***Index Terms—Symmetric Encryption, Asymmetric Encryption, Cryptography, Cipher text, Plain text, Decryption***

## I. INTRODUCTION

Cryptography plays a very vital role in keeping the message safe as the data is in transit. It ensures that the message being sent at one end remains confidential and should be received only by the intended receiver at the other end. Cryptography converts the original message in to non readable format and sends the message over an insecure channel. The people who are unauthorized to read the message try to break the non readable message but it is hard to do it so. The authorized person has the capability to convert the non readable message to readable one.

The original message or the actual message that the person wishes to communicate with the other is defined as Plain Text. The message that cannot be understood by anyone or meaningless message is what we call as Cipher Text. Encryption is the process of converting plaintext into cipher text with a key. A Key is a numeric or alpha numeric text or may be a special symbol. A decryption is a reverse process of encryption in which original message is retrieved from the cipher text. Encryption takes place at the sender end and Decryption takes place at the receiver end.

Figure 1 shows the encryption/decryption process of a plaintext message. The input to the encryption process is plain text and that of decryption process is cipher text. First the plaintext is passed through the encryption algorithm which encrypts the plaintext using a key and then the produced cipher text is transmitted.

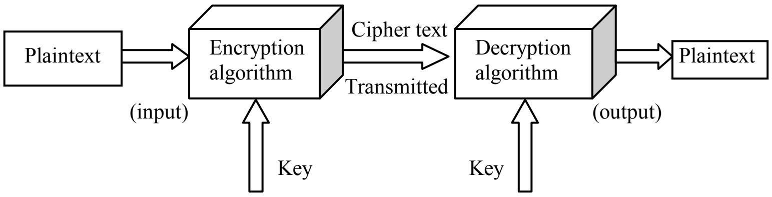

Figure 1: Encryption/Decryption process

At the end of decryption, the input cipher text is passed through the decryption algorithm which decrypts the cipher text using the same key as that of encryption. Finally we get the original plaintext message.

### *A. Goals of Cryptography*

Cryptography provides a number of security goals to ensure the privacy of data, non alteration of data and so on. Due to the great security advantages of cryptography it is widely used today. Following are the various goals of cryptography [1].

- Confidentiality
  Information in computer is transmitted and has to be accessed only by the authorized party.
- Authentication
  The information received by any system has to check the identity of the sender that whether the information is arriving from a authorized person or a false identity.
- Integrity
  Only the authorized party is allowed to modify the transmitted information. No one in between the sender and receiver are allowed to alter the given message.
- Non Repudiation
  Ensures neither the sender, nor the receiver of message can deny the transmission.
- Access Control
  Only the authorized parties are able to access the given information.

### *B. Types of Cryptography*

Encryption algorithms can be classified into two broad categories- Symmetric key Cryptography and Asymmetric Key Cryptography.

- Symmetric Key Cryptography
  In symmetric Cryptography the key used for encryption is similar to the key used in decryption. Thus the key distribution has to be made prior to the transmission of information. The key plays a very important role in symmetric cryptography since their security directly depends on the nature of key i.e. the key length etc. There are various symmetric key algorithms such as DES, TRIPLE DES, AES, RC4, RC6, BLOWFISH [2].
- Asymmetric Key Cryptography
  In Asymmetric Cryptography, two different keys are used for encryption and decryption- Public and Private. The public key is meant for general use so

**Manuscript received on August, 2012**

**Monika Agrawal**, Pursuing MTech, Department Of Computer Science & Engineering, SSCET-Bhilai(C.G), India.

**Pradeep Mishra**, Professor, Department Of Computer Science & Engineering, SSCET- Bhilai(C.G), India.

it is available to anyone on the network. Anyone who wants to encrypt the plaintext should know the Public Key of receiver. Only the authorized person can be able to decrypt the cipher text through his own private key. Private Key is kept secret from the outside world.

Symmetric Encryption Algorithm runs faster as compared to Asymmetric key algorithms. Also the memory requirement of Symmetric algorithm is lesser as compared to asymmetric [3].

## II. REVIEW OF RELATED WORK

Many cryptographic algorithms have already been proposed and implemented to provide security to the user that his/her message would remain safe at the time of communication over the web. But now a days hacking has become a common practice in society which made such cryptographic algorithms no longer safe. In this paper we have studied a number of such symmetric key algorithms and selected one of them for implementation and further enhancement.

- DATA ENCRYPTION STANDARD (DES)

DES (Data Encryption Standard) was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). It was developed by an IBM team around 1974 and adopted as a national standard in 1997.

DES is a 64-bit block cipher under 56-bit key. The algorithm processes with an initial permutation, sixteen rounds block cipher and a final permutation. DES application is very popular in commercial, military, and other domains in the last decades [4].

Although the DES standard is public, the design criteria used are classified. There has been considerable controversy over the design, particularly in the choice of a 56-bit key [5].

- TRIPLE DES (TDES)

The triple DES (3DES) algorithm was needed as a replacement for DES due to advances in key searching [6]. TDES uses three round message. This provides TDES as a strongest encryption algorithm since it is extremely hard to break 2^168 possible combinations.

Another option is to use two different keys for the encryption algorithm. This reduces the memory requirement of keys in TDES. The disadvantage of this algorithm is that it is too time consuming.

- ADVANCED ENCRYPTION STANDARD (AES)

AES was developed by two scientists Joan and Vincent Rijmen in 2000. AES uses the Rijndael block cipher. Rijndael key and block length can be 128, 192 or 256-bits. If both the key-length and block length are 128-bit, Rijndael will perform 9 processing rounds. If the block or key is 192-bit, it performs 11 processing rounds. If either is 256-bit, Rijndael performs 13 processing rounds [7].

Each processing round involves four steps:

1. Substitute bytes – Uses an S-box to perform a byte by byte substitution of the block,
2. Shift rows – A simple permutation,
3. Mix column – A substitution method where data in each column from the shift row step is multiplied by the algorithm's matrix and
4. Add round key – The key for the processing round is XORed with the data.

- BLOWFISH

Bruce Schneier designed blowfish in 1993 as a fast, free alternative to existing encryption algorithms. Since then it has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm.

The Blowfish algorithm has many advantages. It is suitable and efficient for hardware implementation and no license is required [8]. The elementary operators of Blowfish algorithm include table lookup, addition and XOR. The table includes four S-boxes and a P-array. Blowfish is a cipher based on Feistel rounds, and the design of the F-function used amounts to a simplification of the principles used in DES to provide the same security with greater speed and efficiency in software. Blowfish is a 64 bit block cipher and is suggested as a replacement for DES.

Blowfish is a fast algorithm and can encrypt data on 32-bit microprocessors at a rate of one byte every 26 clock cycles. The algorithm is compact and can run in less than 5K of memory [9].

## III. METHODOLOGY

Some specifications of Blowfish algorithm are as follows-

- A 64 bit block cipher with a variable key length.
- There is a P-array and four 32-bit S-boxes. The P-array contains 18 of 32-bit subkeys, while each S-box contains 256 entries.
- The algorithm consists of two parts: a key-expansion part and a data-encryption part.
- Key expansion converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes.
- The data encryption occurs via a 16-round Feistel network. Each round consists of a key-dependent permutation, and a key and data-dependent substitution.
- All operations are XORs and additions on 32-bit words.
- The input is a 64 bit data element.

The process of Subkey generation is illustrated as follows-

1. Initialize P array and S boxes with Hexadecimal digits of Pi.
2. XOR P-array with the key bits (i.e., P1 XOR (first 32 bits of key), P2 XOR (second 32 bits of key)…
3. Use the above method to encrypt the all-zero string.
4. This new output is P1 and P2.
5. Encrypt the new P1 and P2 with the modified subkeys.
6. This new output is now P3 and P4.
7. Repeat the above steps until we get all the elements of P array i.e P1, P2….

The encryption algorithm for Blowfish is illustrated as follows:

**Table I:** Blowfish Encryption Algorithm

```
1. Divide X into two 32-bit halves: XL, XR
2. For i = 1 to 16
       $XL = XL \oplus Pi$
       $XR=F(XL) \oplus XR$
       Swap XL and XR
3. Swap XL and XR (Undo the last swap )
4. $XR=XR \oplus P17$
5. $XL = XL \oplus P18$
6. Concatenate XL and XR
```

**Table 1** shows the algorithm for blowfish encryption. In this algorithm an F function is used which is represented by the Figure 2.

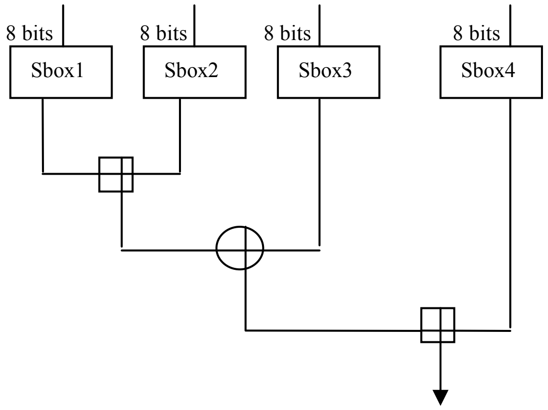

Figure 2: Working of F function

The F function uses the substitution boxes of which there are four, each containing 256 32-bit entries [10]. If the block XL is divided into 8-bit blocks a, b, c and d, the function F(XL) is given by the formula:

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

The decryption process is just reverse of the encryption process.

The modified approach for blowfish algorithm consists of the same specifications as that of Blowfish algorithm except that of a random number defined as Rn.

$Rn = rand(\ )$ (2)

where rand is a function in MATLAB that generates a random number and Rn be any integer .

This random number can be any integer without any limit. We restrict the range of this random number to be between 0 to 65535 (i.e within the range of 2^16).Next consider a variable say Flag. The value of this variable can either be 0 or 1.Initially its value remains 0.Next represent Rn in binary format of 16 bit string from MSB to LSB. The positions in which a '0' is encountered from MSB to LSB then set the variable Flag otherwise remains reset. Also note the position of '0' in the string as a round number of blowfish algorithm. The positions in which the value of variable Flag is reset, then encryption process remains the same as for the blowfish algorithm in that round. But when the positions of the string have value of variable Flag as set, then no F function will be applied in that round which means that value of XL is directly passed for calculation of XR in step 2 of Table 1.

For example, suppose the random number generated is Rn= 29339 ; and Flag=0. The binary representation of Rn as 16 bit string is 0111001010111001

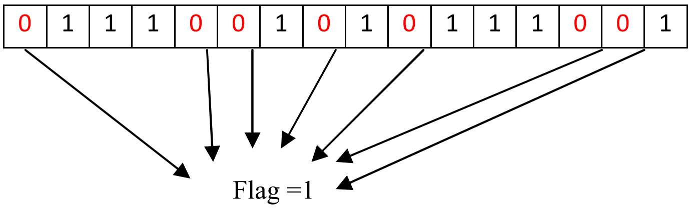

Figure 3 Sample example of working of Modified blowfish encryption algorithm.

In Figure 3 the positions that are holding 0 entries are 1, 5, 6, 8, 10, 14 and 15. In all these positions the value of flag is set otherwise remains '0'. These 0 holding positions are nothing but are round numbers of blowfish encryption algorithm. In such rounds the F function will not work in the statement :

$XR= F(XL) \oplus XR$ (3)

The remaining steps remain the same as that of Blowfish algorithm. The modified blowfish encryption algorithm runs on the input plain text as shown in Table 2.

**Table II:** Modified Blowfish Encryption Algorithm

```
1. Divide X into two 32-bit halves: XL, XR.
2. Generate a random number say Rn and set the
   variable Flag to 0.Represent Rn in form of 16 bit
   binary string say str.
3. For i=1 to 16
          If str[i] == '0'
       Set Flag=1
       If Flag == '1'
               XL = XL ⊕ Pi
               XR=  XL ⊕ XR
       Else
               XL = XL ⊕ Pi
               XR=F(XL) ⊕ XR
       Swap XL and XR
4. Swap XL and XR (Undo the last swap )
5. XR=XR ⊕ P17
6. XL = XL ⊕ P18
7. Concatenate XL and XR
```

## IV. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

The Modified Blowfish algorithm was successfully implemented in MATLAB with an environment Windows 7 Home Basic (64-bit) Operating System, Intel Core i3- 2330M Processor 2.20 GHz clock rate, and Memory of 2GB RAM, 320 GB Hard Disk.

**Table III:** Experimental Results of Encryption time, Decryption time, and Throughput for Blowfish and Modified Blowfish algorithm

| Input File Size (KB) | Encryption Time | | Decryption Time | | Throughput | |
|---|---|---|---|---|---|---|
| | Blow-Fish | Modi fied | Blow-Fish | Modi fied | Blow fish | Modifid |
| 3 | 3.54 | 2.26 | 3.53 | 2.24 | 0.84 | 1.32 |
| 5 | 5.76 | 2.35 | 5.8 | 2.33 | 0.86 | 2.12 |
| 7 | 8.31 | 1.77 | 8.21 | 1.73 | 0.84 | 3.95 |
| 11 | 13.78 | 7.99 | 13.65 | 7.87 | 0.79 | 1.37 |
| 16 | 22.19 | 8.65 | 21.89 | 8.64 | 0.72 | 1.84 |
| 21 | 28.28 | 19.06 | 27.99 | 17.99 | 0.74 | 1.10 |

Table 3 shows a detailed comparison of parameters based on encryption time, decryption time and throughput for the Blowfish and Modified blowfish algorithm. These results have been plotted against different input file sizes (in kilobytes).

The new concept of change in number of times the F Function is applied over the plaintext of 64 bits gives an enormous improvement on two major points. First, the security aspect of Blowfish algorithm has improved because every time a new random number will be generated and it will be unpredictable to know in which iteration the F function will be working and in which the F function will not. The second important point is that as the number of times the F function will be working gets reduced, the time taken for encryption and decryption automatically gets down as compared to the blowfish algorithm. Three Performance Metrics has been considered here-

### *1. Encryption Time*

Encryption Time is one of a performance metric which is defined as the amount of time required for converting plaintext message to cipher text at the time of encryption.

Figure 4.1 shows a graph for comparison of encryption time taking different input file sizes. The blue colored bar represents encryption time with Blowfish algorithm and red colored bar shows encryption time with Modified Blowfish algorithm. It is clear from the graph that the amount of encryption time taken by Modified Blowfish algorithm is almost half as compared to that of Blowfish algorithm for the same input.

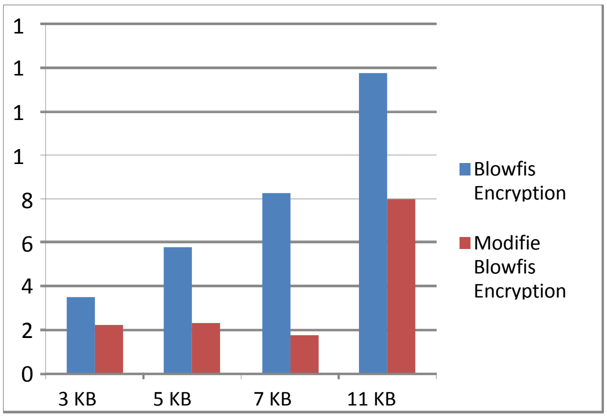

Figure 4.1: Comparison on Encryption time of Blowfish and Modified Blowfish algorithm with different input file size

### 2. Decryption Time

Decryption Time is one of a performance metric which is defined as the amount of time required for converting the cipher text into the plain text at the time of decryption.

Figure 4.2 shows a graph for comparison of Decryption time taking different input file sizes. The blue colored bar represents decryption time with Blowfish algorithm and red colored bar shows decryption time with Modified Blowfish algorithm. It is clear from the graph that the amount of decryption time taken by Modified Blowfish algorithm is almost half as compared to that of Blowfish algorithm for the same input.

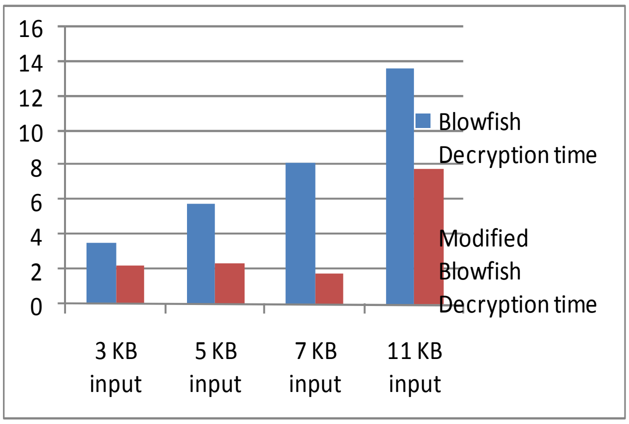

Figure 4.2: Comparison on Decryption time of Blowfish and Modified Blowfish algorithm with different input file sizes

### *3. Throughput*

The throughput of the encryption scheme is calculated by dividing the total plaintext in Megabytes encrypted on the total encryption time for each algorithm in.

$$\text{Throughput}= \frac{\text{Total Plaintext in MegaBytes}}{\text{Encryption Time}}$$

The higher the value of throughput more is the efficiency of encrypting any text with an encryption algorithm. Figure 4.3 shows a graph for comparison on throughput of Blowfish and Modified Blowfish algorithm with different input file sizes. A blue colored line indicates throughput for blowfish algorithm and a red colored line indicates throughput for modified blowfish algorithm. It is clear from the below fig that throughput is higher for Modified blowfish algorithm as compared to Blowfish algorithm.

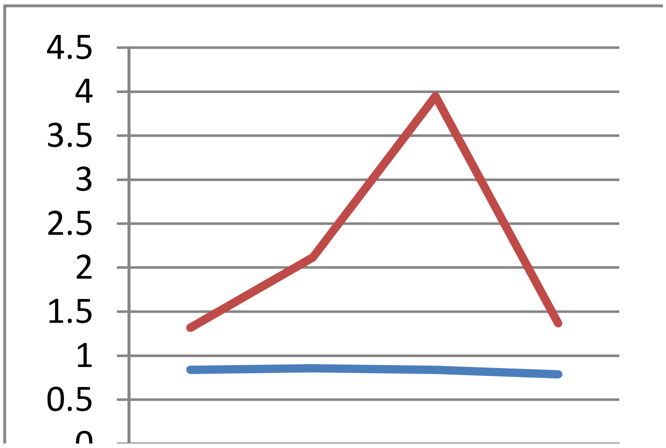

Figure 4.3: Comparison on Throughput of Blowfish and Modified Blowfish with different input file sizes.

## V. CONCLUSION

This paper gives a detailed study of the most popular symmetric key encryption algorithm that is Blowfish and discussed about its advantages. Based on the benefits of Blowfish algorithm we have proposed and implemented a new approach to further enhance the existing algorithm to achieve better results in terms of parameters such as Encryption time, Decryption time and Throughput.

The striking feature of modified blowfish encryption algorithm is that for the same input plaintext the cipher text generated at each time will be different. This is because every time a new random number gets generated and this as a result gives difference in the application of F function over each round. The advantage of different cipher text generated for the same input is it will greatly enhance the security aspect of blowfish algorithm.

The second biggest advantage of this approach is that it is less time consuming as compared to blowfish algorithm. The above results clearly indicate that the encryption time and decryption time for modified blowfish algorithm is almost half to that of blowfish algorithm.

## VI. ACKNOWLEDGMENT

The authors are thankful to Dr. K.K Mehta, Professor and Head Of Computer Science & Engineering Department of Shri Shankaracharya College Of Engineering & Technology, Bhilai (C.G), India for giving thoughtful suggestions during our work that enabled us to present this work.

## REFERENCES

[1] O.P Verma, Ritu Agarwal, Dhiraj Dafouti and Shobha Tyagi, “Peformance Analysis Of Data Encryption Algorithms”,IEEE Delhi Technological University India, 2011.

[2] Diaa Salama, Abdul. Elminaam, Hatem Mohamed, Abdul Kader and Mohie Mohamed Hadhoud, “Performance Evaluation of Symmetric Encryption Algorithms”, International Journal of Computer Science and Network Security, vol.8 No.12, December 2008.

[3] Ketu File white papers, “Symmetric vs Asymmetric Encryption”, a division of Midwest Research Corporation.

[4] “Data Encryption Standard,” Federal Information Processing Standards Publication No. 46, National Bureau of Standards, January 15, 1977.

[5] Tingyuan Nie and Teng Zhang,“ A Study of DES and Blowfish Encryption Algorithm”, IEEE, 2009.

[6] Aamer Nadeem and Dr M. Younus Javed , “A Performance Comparison of Data Encryption Algorithms”, IEEE, 2005.

[7] Himani Agrawal and Monisha Sharma, “Implementation and analysis of various symmetric cryptosystems”, Indian Journal of Science and Technology Vol. 3 No. 12, December 2010.

[8] Allam Mousa, “Data Encryption Performance Based on Blowfish”, 47th International Symposium ELMAR-2005.08-1 0, June 2005.

[9] Noohul Basheer Zain Ali, and James M Noras “OPTIMAL DATAPATH DESIGN FOR A CRYPTOGRAPHIC PROCESSOR: THE BLOWFISH ALGORITHM” Malaysian Journal of Computer Science, Vol. 14 No. 1, June 2001.

[10] Russell K. Meyers and Ahmed H. Desoky “An Implementation of the Blowfish Cryptosystem”, IEEE, 2008.

Ms.Monika Agrawal obtained her B.E (Computers) in 2008 from Bhilai Institute Of Technology, Durg. She is pursuing her M.E in Computer Technology and Applications from Shri Shankaracharya College Of Engineering & Technology (SSCET) ,Bhilai, C.G, India. Presently she is working as an Assistant Professor at Shri Shankaracharya Institute Of Technology & Management (SSITM),Bhilai, C.G., India.

Mr. Pradeep Mishra obtained his M.E (Computer Technology Application) in 2008 from Shri Shankaracharya College Of Engineering & Technology (SSCET). He is pursuing PHD from Technical University Of Chhattisgarh State, Bhilai(C.G), India. Presently, he is working as a Professor at S.S.C.E.T, Bhilai (C.G.), India. He has papers in 4 International Journals, 1 International Conference and 4 National Conferences. His area of interest includes Artificial Intelligence, Cryptography and Network Security, Image Processing.