Evaluation of Blowfish Algorithm based on Avalanche Effect

Manisha S. Mahindrakar
Department of Computer Science and Engineering
Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, Maharashtra, India

Abstract — In the modern era of information technology, many applications are internet based. These need end-to-end information security to ensure data privacy, authentication and integrity. Data transfer in communication network, encryption algorithm plays an important role for providing security of information. The principle goal behind design or evaluation of any encryption algorithm is security against unauthorized access. The evaluation criterion includes security analysis, avalanche effect, encryption/decryption computation time, power consumption, memory requirement and so on. The avalanche effect is an important parameter for evaluation of any cryptographic algorithms. Blowfish is one of the widely used algorithms in wireless network. In this paper, I have evaluated avalanche effect in blowfish algorithm. It has been concluded that Blowfish exhibits strong avalanche effect in each round.

Keywords – Blowfish algorithm, Avalanche effect, Hamming distance

I. INTRODUCTION

In recent years, a lot of applications based on internet such as online-banking, online shopping, stock trading, electronic transactions for bill payments etc. are emerged. Such confidential transactions over wire or wireless public network demand end-to-end secure connections to ensure data privacy, authentication and integrity [1].

Use of encryption algorithm is vital for information security guarantee. Encryption is the process used to change actual data into a format that cannot be recognized by anyone except the sender and receiver.

Encryption algorithms can be categorized into two types: Symmetric key and Asymmetric key encryption. Some well-known examples of Symmetric key encryption algorithm are Data Encryption Standard (DES), 3DES, Advanced Encryption Standard (AES), Rivest Cipher 5 (RC5), blowfish etc. which use certain- or variable- length key. All these algorithms use single key for encryption and decryption. Key distribution is potential problem in use of these algorithms.

Asymmetric encryption algorithm uses two keys - public and private. Public key is used for encryption and private key is used for decryption. Hence these algorithms are more secure and solve the problem of key distribution. The most commonly used algorithm of this kind is Rivest Shamir Adleman (RSA) algorithm. The main disadvantage of this type is that they are based on mathematical functions and are not very efficient for small mobile devices [2].

One of the important considerations for measuring the strength of a cryptographic algorithm is its avalanche effect. Avalanche effect is small change in either the plaintext or the key should produce a significant change in the cipher text. In particular, a change in one bit in the plaintext, for the same key, should invert each bit in cipher text with probability 0.5. Likewise, a change in one bit in the key, for the same plaintext, should invert each bit in cipher text with probability 0.5 [3]. If the change were small this might provide a way to reduce the size of the plaintext or key space to be searched.

Blowfish was one of the first secure block ciphers not subject to any patents and therefore freely available for anyone to use. This benefit has contributed to its popularity in cryptographic software. This algorithm is extensively used in numerous products for security of communication systems like wireless network, portable terminal etc. It is also used in applications like file and disk encryption, password management, archiving tools, backup tools, database security and many more [4]. So it is essential to evaluate its performance to ensure its domain applications.

II. BLOWFISH ALGORITHM

A. Related Work:

As of today, the blowfish algorithm has no known cryptanalysis. Hence this algorithm is gaining acceptance as strong encryption algorithm [5]. This algorithm along with other symmetric ciphers has been analyzed considerably. The security and speed of the algorithm is analyzed, in [6], by applying different type of input data such as text, sound and image. DES and Blowfish algorithms are analyzed, based on encryption speed and power consumption, in [7]. [8] implements and compares DES, 3DES, AES, Blowfish and RC4 based on avalanche effect, memory required for implementation and simulation time required for different messages. In this paper, I have experimented and analyzed the avalanche effect of Blowfish algorithm after each round.

Blowfish is a symmetric variable key-length block cipher designed by Bruce Schneier [4].

B. Characteristics of Blowfish:

- It is variably secure as the key length is variable from 4 bytes to 56 bytes. This allows a tradeoff between higher speed and higher security.
- It is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors (encrypts data at a rate of 18 clock per byte) with large data caches.
- It is compact algorithm as it requires less than 5K of memory.
- Blowfish is easy to implement as it has simple structure.

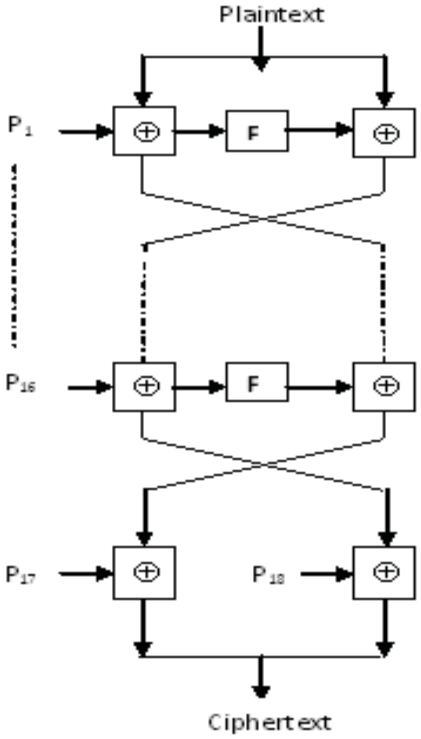

Figure 1. Blowfish encryption algorithm

C. Working of Blowfish:

Blowfish algorithm encrypts 64-bit block of plaintext into 64-bit block of ciphertext using 16 rounds as shown in figure 1 [9]. It works in two steps. In first step, using the key, blowfish initializes 18 P-arrays which contains subkeys of size 32-bits and four S-boxes, each with 256 32-bits entries. A total 521 executions of the blowfish encryption algorithm are required to produce the final S- and P-arrays.

Second step is data encryption which uses P-arrays and S-boxes initialized in the first step. Data encryption comprises 16-round Feistel network. Unlike classical Feistel cipher, Blowfish operates on both halves of data in each round. Unlike other symmetric block ciphers, blowfish decryption occurs in the same algorithmic direction, rather than the reverse. However, like other block ciphers, decryption involves use of subkeys in the reverse order.

Figure 2 shows details of a single round in Blowfish. Each round consists of a key (P_i) dependent permutation, and a key- and data-dependent substitution. The 32-bit input to function **F** is divided into four bytes as shown in the figure 2. If I consider those bytes p, q, r and s then F can be written as follows:

$$F(p,q,r,s) = \left(\left(S_{1,a} + S_{2,b}\right) \oplus S_{3,c}\right) + S_{4,d}$$

Thus, each round includes the complex use of addition modulo 2^{32} and XOR, plus substitution using S-boxes making cryptanalysis difficult.

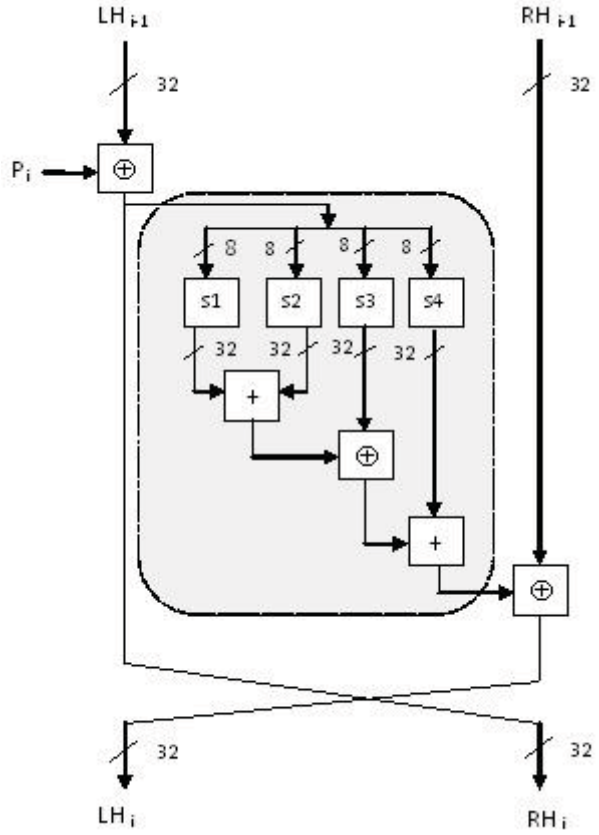

Figure 2: A single Blowfish round

The subkeys and S-Boxes arrays require over 4 Kbytes of memory. Hence this algorithm is not suitable for applications with limited memory, such as smart cards. Also the initialization step is complex, so this algorithm is not suitable for applications in which symmetric key changes frequently.

Blowfish is suitable for applications where the key does not change often, like a communications link or an automatic file encryptor. It is used in a number of implementations, including Secure Shell (SSH) technologies and HDTV transmissions [8].

III. Experiment and Result

I have performed this experiment to analyze avalanche effect in blowfish algorithm. For this purpose, I have designed the experimental setup in Linux operating system. The C source code for blowfish algorithm is taken from [10] and modified to generate the results. I have taken 64-bit input plaintext block size, 64-bit key size and 16 rounds to generate 64-bit output ciphertext block.

Result is analyzed by using two keys that differ in only one bit position and by using two plaintexts that differ in only one bit position. For both of the cases, the hamming distance is calculated to get the number of bits that differ between the two ciphertexts. Then, the avalanche effect is calculated as

$$\text{Avalanche Effect} = \frac{\text{Hamming distance}}{\text{Block Size}} \times 100\ \%$$

Section 4 shows the results of our experiment. The Table column 1 shows the avalanche effect for 16 rounds when I took keys as "ABCDEFGH" and flipping one bit from the key to get "CBCDEFGH" (on flipping A (0100 0001) to C (0100 0011)). The input plaintext used is "security".

The Table column 2 shows the avalanche effect for 16 rounds when I took input plaintext as "security" and flipping one bit from the plaintext to get "secusity" (on flipping r (0111 0010) to s (0111 0011)). The key used is "ABCDEFGH".

The following table shows results after calculating avalanche effect in key change and plaintext change for 1 to 16 rounds of Blowfish algorithm.

Table 1: Avalanche effect

Avalanche effect when change in key by 1-bit			Avalanche effect when change in plaintext by 1-bit		
Round	**Hamming distance**	**Avalanche effect in %**	**Round**	**Hamming distance**	**Avalanche effect in %**
1	18	28	1	1	2
2	33	52	2	16	25
3	27	42	3	30	47
4	29	45	4	30	47
5	30	47	5	31	48
6	26	41	6	26	41
7	33	52	7	29	45
8	32	50	8	33	52
9	31	48	9	31	48
10	27	42	10	30	47
11	41	64	11	31	48
12	32	50	12	35	55
13	38	59	13	32	50
14	37	58	14	31	48
15	23	36	15	33	52
16	26	41	16	27	42

IV.CONCLUSION

So far, the security of blowfish is unchallenged. One of the important factors in its strength is initialization of subkeys and S-boxes. S-boxes are key dependent and both the subkeys and S-boxes are produced by a process of repeated application of blowfish itself. Thus during encryption, in each round, the input bits get thoroughly mixed and resulting in many changes in output bits making cryptanalysis very difficult.

From the experimental results I can clearly see that the avalanche effect exhibited by blowfish algorithm is very strong. Approximately 50% ciphertext bits differ after every round. Also I can see that avalanche effect is stronger when plaintext is changed than the change in key.

REFERENCES

[1] Ramesh Karri and Piyush Mishra, "Minimizing the secure wireless session energy," Journal of Mobile Network and Applications (MONET), vol. 8, Apr. 2002, pp. 177-185.

[2] Diaa Salama Abdul, Elminaam, Hatem Mohamed Abdul Kader and Mohie Mohamed Hadhoud3, "Performance Evaluation of Symmetric Encryption Algorithms," in IJCSNS International Journal of Computer Science and Network Security, vol.8 No 12, December 2008, pp. 280-286

[3] Bernard Menezes, Network Security and Cryptography.

[4] Bruce Schneier "The Blowfish encryption algorithm," http://www.schneier.com/paper-blowfish-fse.html

[5] Strong Encryption, http://www.tropsoft.com/strongenc/blowfish.htm
[6] Allam Mousa, "Data encryption performance based on blowfish," 47th international Symposium ELMAR-2005, pp.131-134,2005.
[7] Tingyuan Nie, Chuanwang Song, Xulong Zhi, "Performance Evaluation of DES and Blowfish Algorithms," 2010 International Conference on Biomedical Engineering and Computer Science, 23-25 April, 2010.
[8] Himani Agrawal and Monisha Sharma, "Implementation and analysis of various symmetric cryptosystems," in IJST Indian Journal of cience and Technology, vol.3 No. 12, December 2010, pp.1173-11
[9] William Stallings. Cryptography and Network Security Principles and Practices (Third Edition).
[10] Bruce Schneier, "Blowfish Source Code by Paul Kocher," http://www.schneier.com/blowfish-download.html