Volume 2, Issue 6, June 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Linux Based of Encryption Quality and Security Valuation of Blowfish Algorithm and its Modified Version using Digital Images

Ranjeet Singh, Madan Kumar
Department of Information Technology,
SRM University, NCR Campus Modinagar, India

Abstract--- There has been a tremendous enhancement in the field of cryptography, which tries to manipulate the plaintext so it becomes unreadable, less prone to hacker and crackers, and again obtain the plaintext back by manipulating this unreadable text and images in some way. In this regard, we have modified one secure algorithm Blowfish [1] which are secret – key block cipher that enhance performance by modifying their function. Now in this paper we want show some results of performance analysis Blowfish and compare it with its modified version to prove that the modification does not violate security requirements. For this, we have considered different aspects of security namely, Encryption quality, Key sensitivity test and Statistical Analysis [4] on software implementation, we have implemented Shell and TCL- tk application to show the differences.

Key words: Encryption, Decryption, Avalanche, key sensitivity, Histogram

1. INTRODUCTION

Blowfish [1] is a variable-length key [1], 64- bit block cipher. The algorithm consists of two parts: a key-expansion part and a data-encryption part. Key expansion converts a key permutation, a key and data dependent substitution. All operations are EX-ORs and addition on 32-bit words. bits into several subkey arrays totaling 4168 bytes.

Expansion part and a data – encryption part. Key expansion converts a key of at most 448. Data encryption occurs via a 16- round Feistel network [3] as shown in Figure 1.1. Each round consists of a key-dependent.

II. Subkeys

Blowfish uses a large number of subkeys [3]. These keys must be precomputed before any data encryption or decryption.

The key array also called P-array consists of 18 32 bit subkeys: P1, P2……..P18

There are four 32-bit S-boxes with 256 entries each:

S1,0, S1,1,……..S1,255:
S2,0, S2,1……...S2,255:
S3,0, S3,1,……..S3,255:
S4,0, S4,1,……..S4,255:

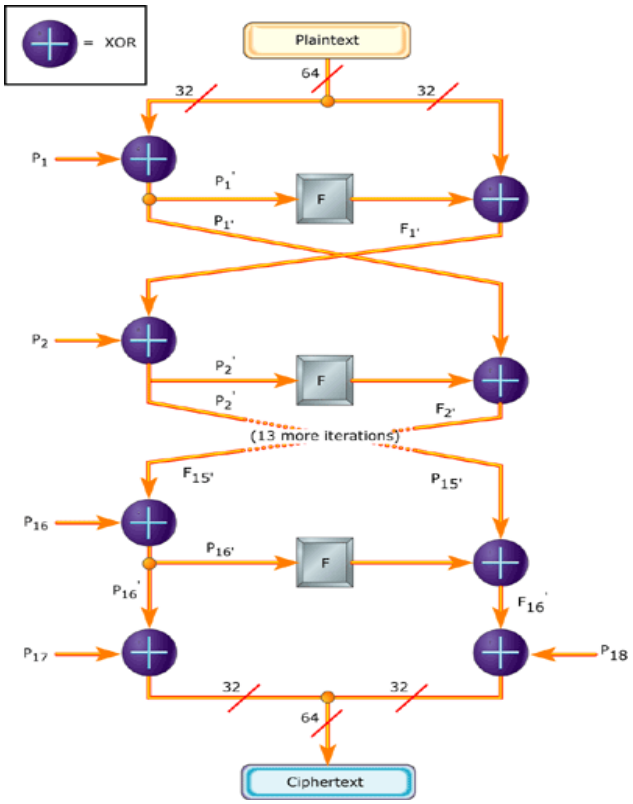

Fig. 1.1 [Blowfish Encryption]

Decryption for Blowfish is relatively straightforward ironically, decryption works in the same algorithmic direction as encryption beginning with the ciphertext as input. However, as expected, the sub-keys are used in reverse order.

Since Function F plays an important role in the algorithm, it was decided to modify function F and determine whether the modified function F saves the time.

Original function F is defined as follows:-

Divide X_L into four eight-bit quarters: a, b, c, d

$$F(XL) = ((S1,a + S2,b \bmod 2^{32}) \text{ XOR } S3,c) + S4,d \bmod 2^{32})$$

Using text vector

Key = "a b c d e f g h I j k l m n o p q r s t u v w x y z";

Plain = "BLOWFISH"

Cipher = 32 4E D0 FE F4 13 A2 03

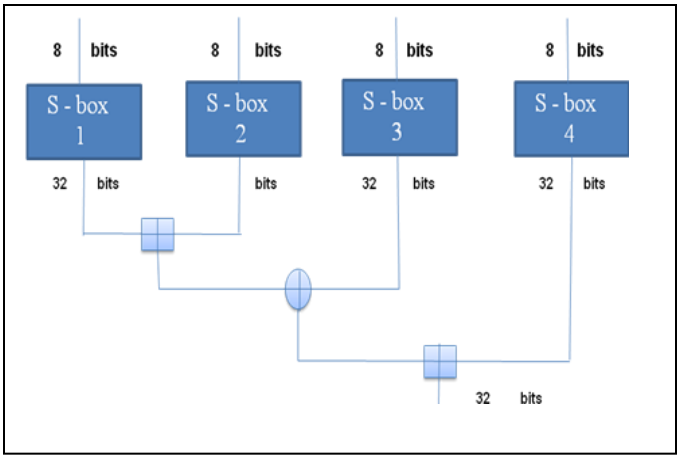

Fig.2. Existing Blowfish Function F

Thus modified Blowfish function F is:

$F(XL) = (S1,a + S2,b \bmod 2^{32}) \text{ XOR } (S3,c + S4,d \bmod 2^{32})$

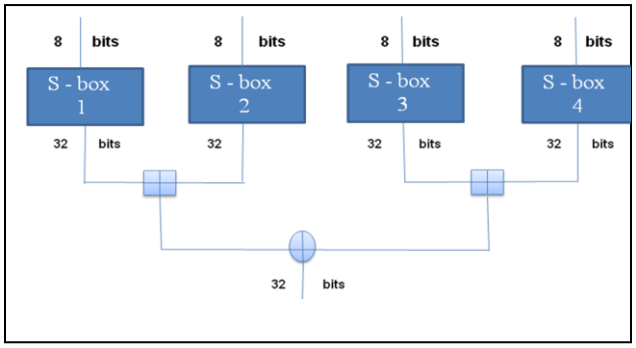

Fig.3. Modified Blowfish Function F

This modification supports the parallel evaluation of two addition operations ($S_{1,a} + S_{2,b} \bmod 2^{32}$) and ($S_{3,c} + S_{4,d} \bmod 2^{32}$) by using timer in Linux C

cpu_time_used = ((double)(ctick_end-ctick_start))/ Clocks_per_Second;

This modification leads to 20% improvement in the execution time of function F.

Now we will show that the above modification does not violate the security of

algorithm. For this, we will make use of avalanche effect, encryption quality, key sensitivity test and statistical analysis.

III. Avalanche Effect

We have used Avalanche effect [1], [2] to show that the modified algorithm also possesses good diffusion characteristics as that of original algorithm.

A desirable feature of any encryption algorithm is that a small change in either the plaintext or the key should produce significant change in the cipher text.

If the change are small, this might provide a way to reduce the size of the plaintext or the key space to be searched and hence makes the cryptanalysis very easy.

We have taken 200 samples each for the original algorithm and modified algorithm and noted down the Avalanche effect by changing the plaintext by one bit between the successive samples.

We have counted the number of times original algorithm gives better avalanche, the number of times modified algorithm give better avalanche, and the number of times both algorithms give same avalanche. Tabulation of results observed by changing one bit of the plaintext in the samples for rounds 2, 4, 6, 8, 10, 12, 14 and 16 of original and modified algorithm.

TABLE 1: Comparison of avalanche effect for Original and modified Blowfish algorithms

No. of Samples	No. of Rounds	Number of times Original algorithm gives better Avalanche	No. of times Modified algorithm gives better Avalanche
200	2	38	46
200	4	41	48
200	6	39	50
200	8	41	52
200	10	37	54
200	12	36	55
200	14	38	49

200	16	41	51

IV. Encryption Quality Analysis

The quality of image encryption [6], [11] may be determined as follows:

Let F and F' denote the original image (plain image) and the encrypted image (cipher image) respectively each of size M*N pixels with L grey levels. F(x, y), F'(x, y) i.e.{0….L-1} are the grey levels of the images F and F' at position (x, y) ($0 \le x \le M-1$, $0 \le y \le N-1$). Let $H_L(F)$ denote the number of occurrences of each grey level L in the original image (plain image) F. Similarly, $H_L(F')$ denotes the number of occurrence of each grey level in the encrypted image (cipher image) F'. The encryption quality represents the average number of changes to each grey level L and is expressed mathematically as

$$\text{Encryption Quality} = \frac{\sum_{L=0}^{255} |H_L(F') - H_L(F)|}{256}$$

The effect of number rounds r on the encryption quality for Blowfish and modified Blowfish is investigated.


Fig.4A Cart.bmp (Original image) **Fig.4B Encrypted using Blowfish Algorithm**

Fig.4C The image after decryption using Blowfish Algorithm


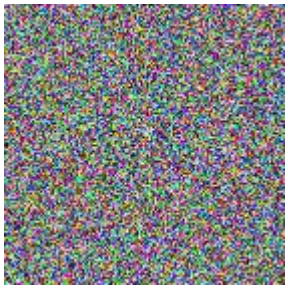

Fig.4D Cart.bmp (Original image) **Fig.4E Encrypted using modified Blowfish Algorithm**

Fig.4F The image after decryption using modified Blowfish Algorithm

The Both images show that modification done to the function does not degrade the quality of encryption.

V. Key Sensitivity Test and Statistical Analysis

We have conducted key sensitivity test[6], [11] on the image butterfly.bmp for original and modified Blowfish algorithms using the following 128 bits keys K1 and K2 where K2 is obtained by complementing one of the 128 bits of K1 which is selected randomly, The hexadecimal digits of K1 and K2 which have this differences bit are shown in bold case.

K1= ADF2**7**856E262AD1F5DEC94A0BF25B27

K2= ADF2**3**856E262AD1F5DEC94A0BF25B27

First the plain image Butterfly.bmp (Fig 5A) is encrypted with K1 using original Blowfish algorithm and then by using K2.These cipher images are shown in Fig. 5B and 5C. Then we have counted the number of pixels that differ in the encrypted images. The result is **99.680687%** of pixels differ from image encrypted with the key K2 from that encrypted with the key K2 from the encrypted with K1.

Above experiment is repeated for modified Blowfish algorithms **99.538920%** of pixels differ from the image encrypted with K1. (Fig.5E) compared to the image encrypted with K2 (Fig. 5F) shows the difference of the two images encrypted with K1 and k2 .

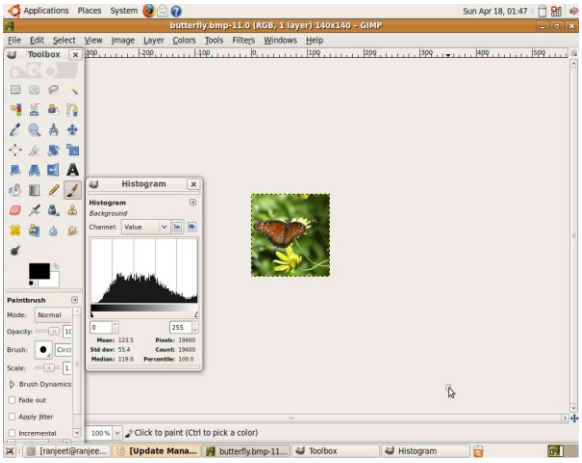

Fig. 5A Original image (butterfly.bmp)

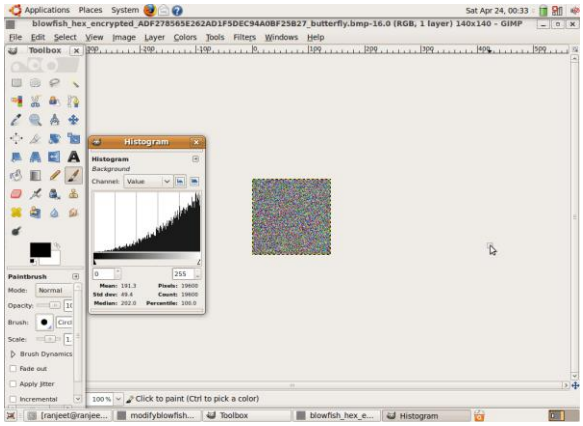

Fig.5B Encrypted with Key K1 using Blowfish algorithm

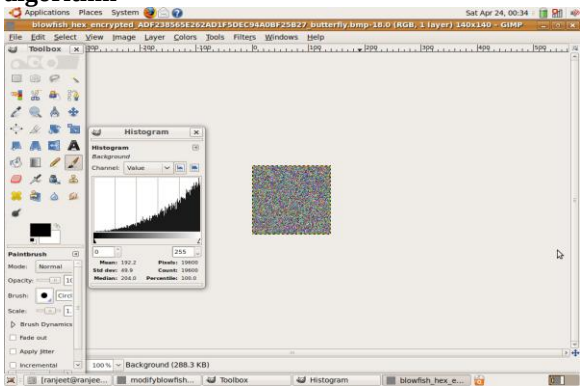

Fig.5C Encypted with Key K2 using Blowfish algorithm

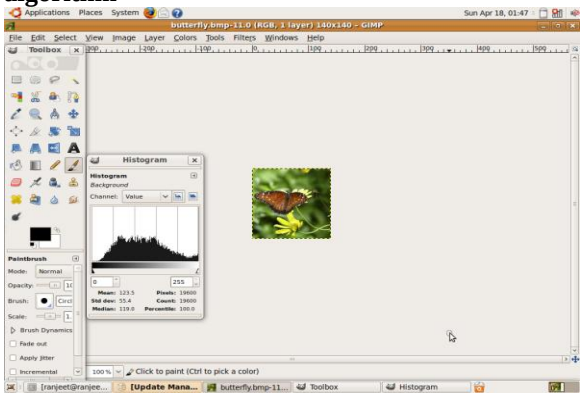

Fig. 5D Original image (butterfly.bmp)

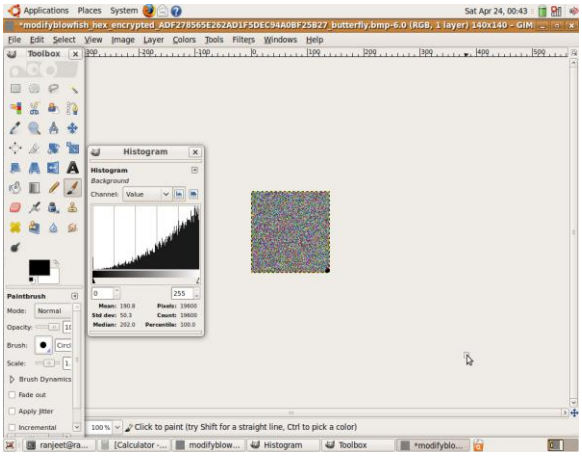

Fig.5E Encrypted with Key K1 using modified Blowfish algorithm

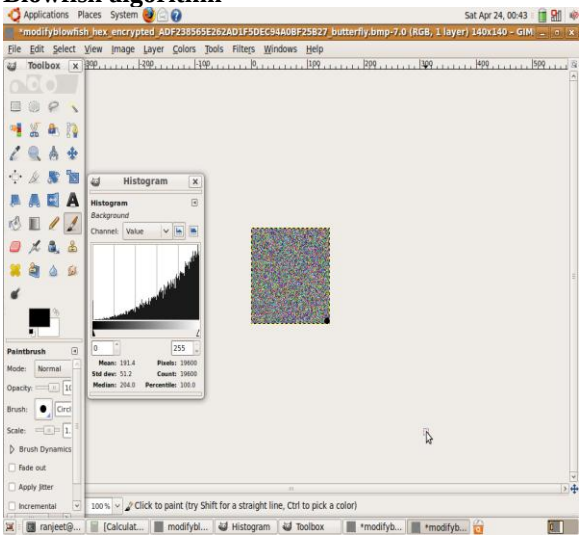

Fig.5F Encrypted with Key K2 using modified Blowfish algorithm

All the results show of Key Sensitivity Test for Blowfish and Modified Blowfish Algorithm. The textures visible in the cipher images of the above tests are an indication of appearance of large area in the original image where pixel values rarely differ. It is the property of block ciphers that for given input there will be fixed cipher text, which means as long as plaintext block repeats, cipher text block also repeats. This can be avoided by using one the modes of operation other than ECB mode.

VI Conclusion

TABLE 2: Comparison of Butterfly .bmp image for Original and modified Blowfish algorithms

Algorithms	Image	Key	Mean	Std Deriv ation
Blowfish	Butterfly.bmp		123.5	55.4

	Encrypted K1	ADF27856E262AD1F 5DEC94A0BF25B27	**19 1.3**	**49.4**
	Encrypted K2	**ADF23856E262AD1F 5DEC94A0BF25B27**	**19 2.2**	**49.9**
MBlowfish	**Encrypted K1**	ADF27856E262AD1F 5DEC94A0BF25B27	**19 0.8**	**50.3**
	Encrypted K2	**ADF23856E262AD1F 5DEC94A0BF25B27**	**19 1.4**	**51.2**

The improved modified algorithm has enhanced the performance over existing blowfish algorithm by reducing standard deviation and mean required for the execution of Blowfish function by 99.5389% and hence increasing the overall execution mean and standard deviation of the modified Blowfish algorithm by 99.6806%. This is explained in detail in the table along with sample images. We have demonstrated that change in one bit in the plaintext produces strong avalanche effect. Hence security of modified algorithm is at least as strong as the original algorithm. We are now trying to theoretically prove this fact. Also we are studying the effects when one bit of the key is changed. Using TCL-TK implementation, it is observed that the reduction.

We have made an attempt to analyze the security of original and modified versions of Blowfish algorithm. We have also tried to demonstrate that the modification made to the function violate the security and is at least as strong as the original algorithm. For this purpose, we have used avalanche criterion, encryption quality, histogram analysis, key sensitivity test and correlation coefficient.

REFERENCES

[1] B. Schneier, *"Applied Cryptography – Protocols, algorithms, and source code in C"*, John Wiley & Sons, Inc., New York, second edition, 1996.

[2] *B. Schneier*, "Description o f a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)", *Fast Software Encryption, Cambridge Security Workshop roceedings (December1993)*, Springer-Verlag, 1994, pp. 191-204.

[3] W. Stallings, Cryptography and Network Security: Principles and Practices, 2nd ed., Prentice Hall, 1999.

[4] Krishnamurthy G.N, Dr. V.Ramaswamy and Mrs. Leela G.H "Performance Enhancement of Blowfish algorithm by modifying its function" Proceedings of International Conference on Computers, Information, System Sciences and Engineering 2006, University of Bridgeport, Bridgeport, CT, USA. pp. 240-244

[5] Harley R. Myler and Arthur R. Weeks, *"The Pocket Handbook of Image Processing Algorithms in C"*,Prentice-Hall, New Jersey, 1993.

[6] *Hossam El-din H. Ahmed, Hamdy M. Kalash, and Osama S. Farag Allah, "Encryption Quality Analysis of RC5 Block Cipher Algorithm for Digital Images",* Journal of Optical Engineering, vol. 45, 2006.

[7] Krishnamurthy G N, Dr. V Ramaswamy "Blow-CAST-Fish, a New 64-bit Block Cipher", IJCSNS, ISSN: 1738-7906, Vol. 8, No.4, pp 282-290, April -2008, Korea.

[8] *B.Schneier, "The Blowfish Encryption Algorithm"*, In *Dr Dobb's Journal*, pp. 38-40, April 1994.

[9] Osama S. Farag Allah, Abdul Hamid M. Ragib, and Nabil A. Ismali, *"Enhancements and Implementation of RC6 Block Cipher for Data Security"*, IEEE Catalog Number:
01CH37239, Published 2001.

[10] Hossam El-din H. Ahmed, Hamdy M. Kalash, and Osama S. Farag Allah, *"An Efficient Chaos-Based Feedback Stream cipher (ECBFSC) for Image Encryption and Decryption",* Accepted for publication in An International Journal of Computing and Informatics, 2007.

[11] *Hossam El-din H. Ahmed, Hamdy M. Kalash. And Osama S. Farang Allah, "Encryption Efficiency Analysis and Security Evaluation of RC6 Block Cipher for Digital Images*", International Journal Of Computer, Information, and System Science, and Engineering volume 1 number 1 2007 ISSN 1307-2331. pp 33-38.

