ENSEMBLE OF BLOWFISH WITH CHAOS BASED S BOX DESIGN FOR TEXT AND IMAGE ENCRYPTION

Jeyamala Chandrasekaran[1] Subramanyan B[1] and Raman G.S[2]

[1]Department of Information Technology, Thiagarajar College of Engineering, Madurai
jeyamala@tce.edu
[2]Department of Information Technology, KLN College of Information Technology, Madurai

ABSTRACT

The rapid and extensive usage of Internet in the present decade has put forth information security as an utmost concern. Most of the commercial transactions taking place over the Internet involves a wide variety of data including text, images, audio and video. With the increasing use of digital techniques for transmitting and storing Multimedia data, the fundamental issue of protecting the confidentiality, integrity and authenticity of the information poses a major challenge for security professionals and hassled to the major developments in Cryptography . In cryptography, an S-Box (Substitution-box) is a basic component of symmetric key algorithms, which performs substitution and is typically used to make the relationship between the key and the cipher text non linear and most of the symmetric key algorithms like DES, Blowfish makes use of S boxes. This paper proposes a new method for design of S boxes based on chaos theory. Chaotic equations are popularly known for its randomness, extreme sensitivity to initial conditions and ergodicity. The modified design has been tested with blowfish algorithm which has no effective crypt analysis reported against its design till date because of its salient design features including the key dependant s boxes and complex key generation process. However every new key requires pre-processing equivalent to encrypting about 4 kilobytes of text, which is very slow compared to other block ciphers and it prevents its usage in memory limited applications and embedded systems. The modified design of S boxes maintains the non linearity [3] [5] and key dependency factors of S boxes with a major reduction in time complexity of generation of S boxes and P arrays. The algorithm has been implemented and the proposed design has been analyzed for size of key space, key sensitivity and Avalanche effect. Experimental results on text and Image Encryption show that the modified design of key generation continues to offer the same level of security as the original Blowfish cipher with a less computational overhead in key generation.

KEYWORDS

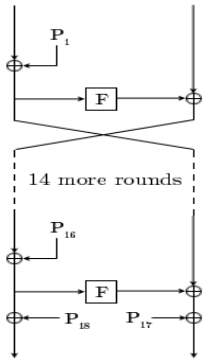
S Box, Chaos, Non Linearity, Symmetric Cryptosystems, Blowfish, Image Encryption

1. INTRODUCTION

An S box can be thought of as a miniature substitution cipher. The input to an s box could be a n bit word, but the output can be an m bit word where m and n are not necessarily the same. An S box can be keyed or keyless and linear or non-linear. Shannon suggested that all block ciphers should have two important properties namely diffusion and confusion. The idea of diffusion is to hide the relation ship between the plain text and the cipher text, which will frustrate the adversary who uses cipher text statistics to find the plain text. Diffusion implies that each symbol in the cipher text is dependant on some or all symbols in the plain text. The idea of confusion is to hide the relation ship between the cipher text and the key, which will frustrate the adversary who uses cipher text to find the key. In other words, if a single bit in the key is changed, most or all bits in the cipher text will also be changed. Every iteration makes use of S boxes, P Boxes, and other non linear operations in order to provide diffusion and confusion.

2. BLOWFISH – ALGORITHM DESCRIPTION

The description of Blowfish algorithm has been referred in [1].Blowfish has a 64-bit block size and a variable key length from 32 up to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes which is similar in structure to CAST-128, which uses fixed S-boxes. There is no effective cryptanalysis on the full-round version of Blowfish known publicly as of 2009. A sign extension bug in one publication of C code has been identified. In 1996, Serge Vaudenay [6] found a known-plaintext attack requiring 2^{8r+1} known plaintexts to break, where r is the number of rounds. Moreover, he also found a class of weak keys that can be detected and broken by the same attack with only 2^{4r+1} known plaintexts. This attack cannot be used against the regular Blowfish; it assumes knowledge of the key-dependent S-boxes. Vincent Rijmen, in his Ph.D. thesis, introduced a second-order differential attack that can break four rounds and no more. There remains no known way to break the full 16 rounds, apart from a brute-force search. Blowfish is a variable-length key, 64-bit block cipher. 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 sub key arrays totaling 4168 bytes. 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 only additional operations are four indexed array data lookups per round.



2.1. Initialization

Blowfish uses a large number of subkeys.

1. The P-array consists of 18 32-bit subkeys:
   P1, P2,..., P18.

2. 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.

2.2. Sub Key Generation

The sub keys are calculated using the Blowfish algorithm. The exact method is as follows:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3). For example:

P1 = 0x243f6a88

P2 = 0x85a308d3

P3 = 0x13198a2e

P4 = 0x03707344

2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)

3. Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2).

4. Replace P1 and P2 with the output of step (3).

5. Encrypt the output of step (3) using the Blowfish algorithm with the modified sub keys.

6. Replace P3 and P4 with the output of step (5).

7. Continue the process, replacing all entries of the P- array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm.

In total, 521 iterations are required to generate all required sub keys. Applications can store the sub keys rather than execute this derivation process multiple times.

3. PROPOSED KEY GENERATION DESIGN

Blowfish algorithm [6] requires 521 encryptions of itself to generate the sub keys namely 18 entries in P arrays and 1024 entries in S boxes. The proposed design replaces the 521 encryptions by adapting chaos functions in order to generate highly non linear and key dependant P arrays and S boxes.

3.1 Chaos Functions

One of the simplest chaos functions is $f(x)=p*x*(1-x)$which is bounded for the limits $0<p<4$. This function can be written in iterative form as$x_{n+1}=p*x_n*(1-x_n)$ with x_0 as the starting value. A thorough treatment and analysis of this function can be found in [3].

3.2. Initial Secret Parameter Exchange

For application of the above function for generating P arrays and S boxes in Blow fish algorithm, it is proposed that the values yielded by the chaos function are to be converted to appropriate key representations. For this, the following three factors have to be agreed upon by the users.

1. The starting value for the iterations (x_0).
2. The number for decimal places of the mantissa that are to be supported by the calculating machine.
3. The number of iterations after which the first value can be picked for generating keys.
4. The number of iterations to be maintained between two picked values thereafter.

3.3. Global Parameters

An indexed key table consisting of all possible keys for a desired key length is published globally. For example if a key of length 32 is required, all possibilities of the keys are generated, tabulated and indexed. The index identifies the key to be selected at any instant of time based on the chaotic value generated.

3.4. Key Generation

Using the above-mentioned secret parameters, both the sender and receiver runs the chaotic equation

$$f(x+1) = 4*x*(1-x)$$

for required number of iterations. The value of every element in P array and S box is identified by the value of the chaotic equation generated during the agreed iteration count. The chaotic value generated will be converted to a suitable index within the range by the formula

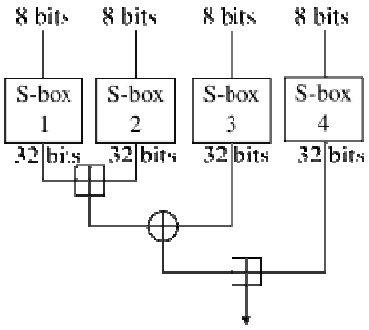
$$z_j = (c_i - f_{min}) * ((index_{max} - index_{min})/(f_{max} - f_{min})) + index_{min}$$

where, z_j is the index to be identified for generating the element in P array or S box, c_i is the chaotic value generated during i^{th} iteration , f_{max} and f_{min} are the maximum and minimum values generated by the chaotic equation, index $_{max}$ and index $_{min}$ are the maximum and minimum values of the indices in key index table.

3.4. Encryption Methodology

We have adopted the Blowfish methodology which is a Feistel network consisting of 16 rounds. The input is a 64-bit data element, x.

Divide x into two 32-bit halves: xL, xR
For i = 1 to 16:
xL = xL XOR Pi
xR = F(xL) XOR xR
Swap xL and xR
Next i
Swap xL and xR (Undo the last swap.)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR
Function F (see Figure 2):
Divide xL into four eight-bit quarters: a, b, c, and d
F(xL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4,d mod 2^{32}



4. EXPERIMENTAL RESULTS

4.1 Text Encryption – Avalenche Effect Analysis

The resistance of any encryption algorithm offered against cryptanalysis is measured by its avalanche effect [5] i.e., For a small change even in one bit of plain text or key should produce significantly differing cipher texts by many number of bits. The modified Blowfish algorithm is experimentally found to exhibit good avalanche effect for plain texts differing by one bit. The results of every round have been tabulated below.

Table 1 Round ciphers for seed differing by order of 10^{-10}

Round	Seed1=0.3	Seed 2=0.30000000001
1	40F6CB26BBE4776A	9BB99294B01BBD8E
2	2652FE3338F04A0B	218A90959BBB3291
3	43B1848AF40D9DC5	F06B6704DE8E5461
4	E46F1E6847AF2AA	FD8DABEB9D669997
5	62C6F5248A4A6BCC	7AB0BA1FD2EB559
6	15687B1D6E4C2941	173A04AF4BE303AF
...	...	...
16	776E8A9FB8F75872	4AA90ED7A7F496C6

4.2 Key Space Analysis

The strength of any cryptographic algorithm depends on the size of its key space to make brute force attack infeasible. The values of P arrays and S boxes depend on the initial seed and its related parameters associated with the chaotic equation. The number of initial seed can vary between 0.2 to 0.8 and hence the key space depend upon the number for decimal places of the mantissa that are to be supported by the calculating machine which is approximately infinitely large making brute force attack computationally infeasible. Even for the same initial seed with different skip value the P arrays and S boxes generated are completely different and non linear.

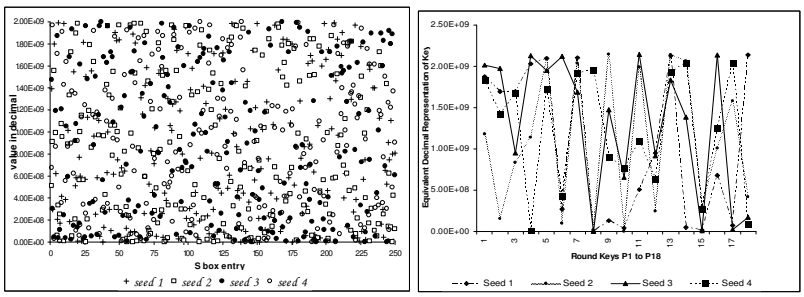


Figure Non Linear Relation Ship between S box and Random Seed Samples

4.3 Key Sensitivity Analysis

The modified Blowfish algorithm is experimented with different seed values for the chaotic equation and the results demonstrate that a perfect non-linear relation ship exists between the plain text and the seed thus providing high resistance towards differential crypt analysis.

Table 2 Key Sensitivity Analysis

Sample	Initial seed Value	Cipher
Seed 1	0.43728423	A8A81271EC3F92D0
Seed 2	0.43728423008	1F26570DBCEFC409
Seed 3	0.43728422999	1DEEDFBD12A21B26
Seed 4	0.4372846578	9F2E0B32FA1A1D5D

4.4 Image Encryption

The algorithm has been implemented in Mat Lab 6.0 in windows environment with a system configuration of PIV processor with 1 GB RAM. The proposed algorithm has been tested with various images in USC-SIPI repository which is a collection of digitized images primarily to support image processing, image analysis and machine vision. (http://sipi.usc.edu/database/).

Figure Original and Encrypted Samples

4.5. Statistical Analysis

Analyzing its strength against statistical attacks proves the robustness of any algorithm. The strength of the proposed algorithm is measured by the histogram analysis of the plain and encrypted images and by the correlation coefficient between the adjacent pixels in both plain and encrypted images.

4.5.1. Histogram Analysis

To prevent the leakage of information to an opponent, it is also advantageous if the cipher image bears little or no statistical similarity to the plain image. An image histogram illustrates how pixels in an image are distributed by graphing the number of pixels at each color intensity level. The histogram of the encrypted image is fairly uniform and significantly different from the respective histograms of the original image and hence does not provide any clue to employ any statistical attack on the proposed image encryption. Figure 3 shows the histogram analysis of plain and original images. The histogram analysis shows that the histogram of the cipher image is fairly uniform and is significantly different from the original image. The encryption algorithm has

covered up all the characters of the plain image and has complicated the statistical relation ship between the plain image and its ciphered version.

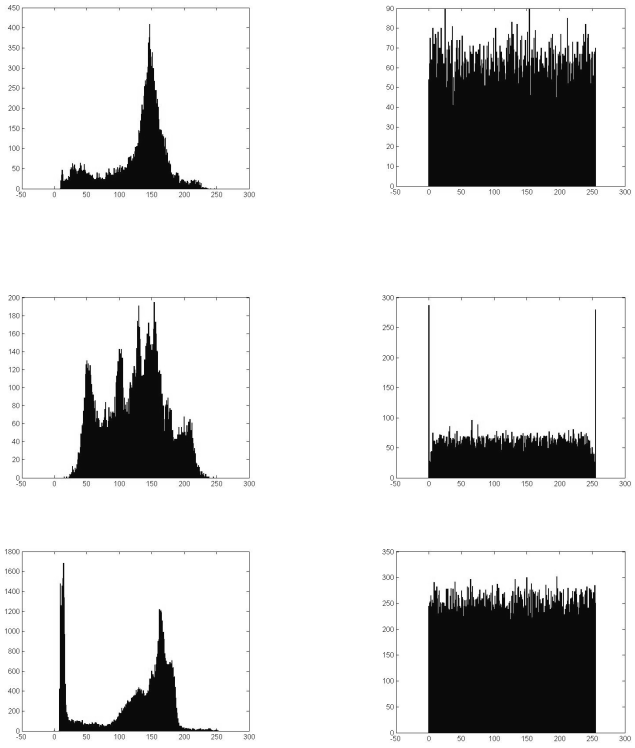


Figure Histograms of original and Encrypted Samples

4.5.2 Correlation Coefficient Analysis

In addition to the histogram analysis, we have also analyzed the correlation between two vertically adjacent pixels, two horizontally adjacent pixels and two diagonally adjacent pixels in plain image/cipher image respectively. The correlation Coefficient is calculated using the formula

$$c.c = \frac{(N * \sum_{j=1}^{N} x_j * y_j - \sum_{j=1}^{N} x_j * \sum_{j=1}^{N} y_j)}{\sqrt{(N * \sum x_j^{\,2} - (\sum_{j=1}^{N} x_j)^2) * (N * \sum y_j^{\,2} - (\sum_{j=1}^{N} y_j)^2}}$$

Where x and y are the values of two adjacent pixels in the image. The figure shows the correlation distribution of two horizontally adjacent pixels in plain image/cipher image for the proposed algorithm. It is clear from the and Table 1 that there is negligible correlation between the two adjacent pixels in the cipher image. However, the two adjacent pixels in the plain image are highly correlated.

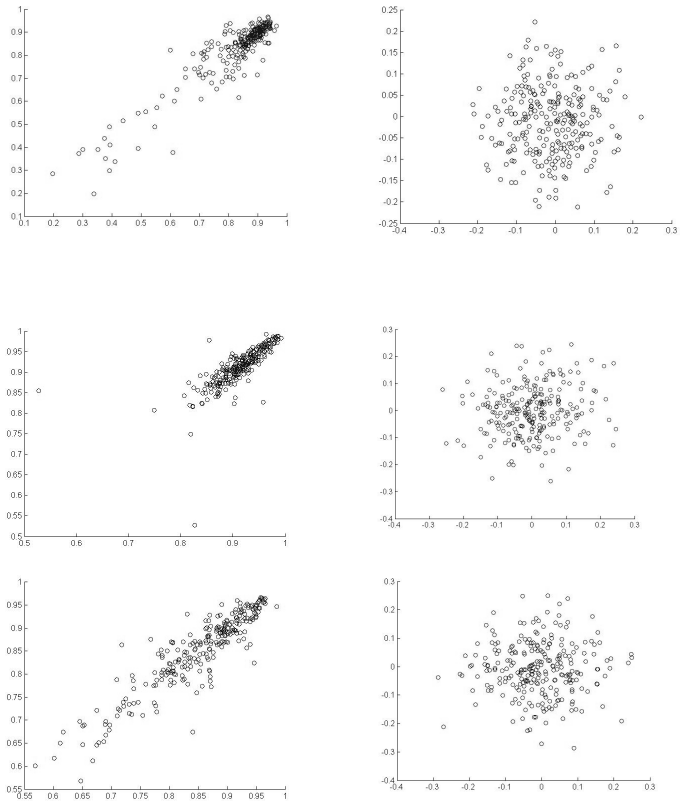


Figure Correlation Between original and Encrypted samples

Table 1. Correlation Between pixels in original and Cipher Images

Image	Original Image	Cipher Image
Ship	0.9004	-0.0014
Lena	0.9510	-0.0080
Camera	0.9565	0.0013

Algorithm	Lena	Gold Hill
Yen, Guo et al	0.0073	0.0106
Socek *et al.*	0.0044	0.0098
Bourbakis et al	1072e-4	1.93e-4
Proposed Algorithm	7.0284e-004	-0.0080

6. CONCLUSIONS

The proposed work explains a new way of generating the elements in and P arrays and S box. Experimental results clearly show that the algorithm generates highly non linear S boxes and P arrays while preserving the same level of security as in Blowfish. The encryption quality for text has been measured by means of key sensitivity tests, key space analysis and Avalenche effect analysis. The algorithm produces good quality cipher texts by employing chaos theory for generation of P arrays and S boxes used in blowfish at a reduced time complexity Based on the experimental results for image Encryption, it can be observed that the proposed cipher offers high encryption quality with minimal memory requirement and computational time. Also the algorithm

offers sufficient resistance towards Brute force attack and statistical crypt analysis of original and encrypted images.

REFERENCES

[1] www.schneier.com/paper-blowfish-fse.html.

[2] Daemen J R Govaerts and J. Vandewalle. (1994). "Correlation Matrices. Fast Software Encryption." Lecture Notes in Computer Science (LNCS) 1008. Springer-Verlag. 275-285.

[3] Youssef, A., S. Tavares, S. Mister and C. Adams. (1995) "Linear Approximation of Injective S-boxes." *IEEE Electronics Letters.* 31(25): 2168-2169.

[4] Fridrich Jiri, "Symmetric ciphers based on two dimensional chaotic maps", (1998) International . Journal of Bifurcation and Chaos 8 (6), pp. 259–1284

[5] Zhang, X. and Y. Zheng.. (1995) GAC -- the Criterion for Global Avalanche Characteristics of Cryptographic Functions. Journal for Universal Computer Science. 1(5): 316-333.

[6] N.K.Pareek, Vinod Patidar, K.K.Sud,(2005) Cryptography using multiple one dimensional chaotic maps, Communications and Nonlinear Science Numerical simulation 10(7) pp 715-723

[7] N.K.Pareek, Vinod Patidar, K.K.Sud (2006)Image Encryption using chaotic logistic map image and Vision Computing ,24 pp 926 -934.

[8] Mitra, Y. V. Subba Rao, and S. R. M. Prasanna,(2006) A new image encryption approach using combinational permutation techniques, International Journal of Computer Science, vol. 1, no. 2 , pp. 1306- 4428

[9] Socek, S. Li, S. S. Magliveras, and B. Furht, (2005) Enhanced 1-D chaotic key-based algorithm for image encryption, IEEE/CreateNet SecureComm, pp. 406-408, September 5-9, 2005.

[10] Shujun Li, Guanrong Chen and Xuan Zheng, (2004)Chaos-based encryption for digital images and videos, chapter 4 in Multimedia Security Handbook, February 2004.

[11] J.C Yen, J.I Guo, (2000)A new chaotic key based design for image encryption and decryption, Proceedings of the IEEE international symposium circuits and systems, Vol 4,pp 49-52

[12] Marwa Abd El-Wahed, Saleh Mesbah, and Amin Shoukry (2008) Efficiency and Security of some Image Encryption Algorithms ,Proceedings of the World Congress on Engineering Vol I WCE.