ISSN No. 0976-5697

Volume 4, No. 11, Nov-Dec 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Effect of Different Chaotic Maps on a Chaotic Block Cipher Algorithm for Image Cryptosystems

Vishnu G. Kamat
M Tech in Information Security and Management
Department of Computer Science, DIT University
kamatvishnu14@gmail.com

Madhu Sharma
Assistant Professor
Department of Computer Science, DIT University
madhuashishsharma@gmail.com

Abstract: In this paper, we provide analysis details of the algorithm set forth by Mohamed Amin et al. [Commun. Nonlinear Sci. Numer. Simulat. 15 (2010) 3484-97]. They proposed a block cipher algorithm using chaotic Tent map. For our analysis, besides the Tent map we have used 3 different maps viz. Hénon map, Gauss iterated map and Tinkerbell Map individually. We have provided experimentation results of statistical and differential analysis of the algorithm using the said chaotic maps.

Keywords: Encryption; Key Generation; Security Analysis; Chaotic Maps; Block Cipher

I. INTRODUCTION

Information Security is a major concern in almost all fields using the computer for information storage and transmission. This led to the widespread use of cryptography to protect the information. In cryptography, encryption is the task of information encoding, in a way that unauthorized parties cannot view the actual information, but only authorized parties can. The information is changed to an unreadable form (ciphertext), using an encryption algorithm. This is usually done by using an encryption key, which dictates how the message is to be encoded. An authorized party is able to change the message back to readable form by using a decryption algorithm, which uses a decryption key. An encryption/decryption mechanism usually also needs a key-generation algorithm to randomize the keys produced. For basic cryptographic terminology readers should refer to [1],[2].

The traditional algorithms are not very suited to the encryption of image due to the fact that image stores a bulk of redundant data. Also image is made up of pixels which show high correlation among adjacent values. Hence there has been a lot of research done recently for developing image encryption algorithms. Among these is a field of study which uses chaotic maps for the encryption process [3-10].

The chaotic maps display cryptographic properties, such as sensitivity to initial parameters (in cryptographic terms, avalanche effect), ergodicity and random like behavior. These maps generate values which appear random to an observer, but can be regenerated based on the knowledge of system parameters and initial values.

The rest of the paper is as follows: Section 2 provides a brief overview of each map used. Section 3 gives an account of the algorithmic assumptions and initial parameters used for the analysis. The results of the same are shown in section 4. Lastly, Section 5 concludes the paper.

II. A BRIEF OVERVIEW OF CHAOTIC MAPS USED

Given below is a brief introduction of the different maps used in our analysis.

A. *Tent Map:*

Tent map uses a simple chaotic function

$$F(x) = \begin{cases} \alpha x & x < 0.5 \\ \alpha(1-x) & x \geq 0.5 \end{cases} \quad (1)$$

where $0 \leq \alpha \leq 2$; and $x \in [0,1]$. An iterative map is defined by

$$X_{n+1} = F(X_n) \quad (2)$$

The map derives its name due to the tent-like shape of the graph of F(x). Depending on the value of α, the tent map shows a range of dynamic behavior from predictable to chaotic. Chaotic behavior is achieved when $\alpha \in [1.4, 2)$.

B. *Hénon map:*

The Hénon map is a 2D chaotic map. It takes a point (Xn,Yn) and maps it to a new point using the function

$$X_{n+1} = Y_n + 1 - \alpha X_n^2$$
$$Y_{n+1} = \beta X_n \quad (3)$$

The map was introduced by Michel Hénon. The map depends on the parameters- α and β. The Hénon map displays chaotic properties for the values of α=1.4 and β=0.3. For other values of α and β the map may or may not be chaotic.

C. *Gauss iterated map:*

In mathematics, the Gauss map is a nonlinear iterated map given by the Gaussian function:

$$X_{n+1} = \exp(-\alpha X_n^2) + \beta \quad (4)$$

α and β are real parameters. In the parameter real space Xn can be chaotic.

D. *Tinkerbell map:*

The origin of the name of this map is uncertain. The Tinkerbell map is defined by the function

$$X_{n+1} = X^2n - Y^2n + \alpha Xn + \beta Yn;$$
$$Y_{n+1} = 2XnYn + \gamma Xn + \delta Yn \quad (5)$$

Some commonly used values of α, β, γ, and δ are:

$\alpha = 0.9, \beta = -0.6013, \gamma = 2.0, \delta = 0.50$

$\alpha = 0.3, \beta = 0.6000, \gamma = 2.0, \delta = 0.27$

III. PROPOSED METHOD

We have used the encryption algorithm proposed in [3]; but we would like to briefly explain certain assumptions and implementation details as used for the analysis.

We have not used any user defined key. The key array is generated by using the different chaotic maps only. The number of key bytes `t' depends on the rounds i.e. t=4r+8; where r is the number of rounds.

The processing is done on 256 bits (32 bytes) of data at a time using eight 32-bit registers. Since it is a block encryption algorithm, padding is added so as to make the input block size of 32 bytes when the image size is not an integral multiple of 32. A padding of zeros (0-31 bytes) is appended to the end of each row to make each row have enough bytes (multiple of 32) for the encryption algorithm to use per block per row.

For example if the image is of dimensions 254 x 254 pixels, a 2 byte padding of all zeros is appended at the end of each row. The last byte of the image then stores the number of bytes used as padding as a pixel value. This is used to remove the padding after decryption. After retrieving the number of bytes padded 'n', all but last row is checked to determine if zeros exist in all the last n bytes and in n-1 bytes of the last row. The padding is then removed to generate the original image.

We have worked on six 256 x 256 gray scale images as shown in Fig. 1(a-f).

We changed one single pixel of the images (first pixel) to generate new images to test against differential attack. Twelve rounds are used for the encryption. CBC mode of operation is used. The usage of the maps to generate the key array is described in the following subsections.

Figure 1. Plain images (clockwise from top left): (a) Lena (b) Cameraman (c) Mandrill (d) Boats (e) Picard (f) Lonesome.

E. Tent Map:

The valu e of $\alpha = 1.889$ is used in equation (1). The starting value used is $X_0 = 0.8$.

a. A temporary key array k[4r+8] is generated based on the number of rounds 'r' using tent map function.
b. The values ranging from [0-1] are then mapped to numbers in the range [0-255].
c. Then a decimal number `d' is generated by performing bitwise xor of the middle 8 bytes of temporary key array mapped to decimal values.

```
n=r+(r+1); // start of middle 8 bits
d=bitxor(k(n),k(n+1), . . . ,k(n+7));
```

d. The first value of the temporary key array is iterated 'd' number of times to generate the first value of the actual key array.

```
key arr(1)=tent(k(1),t)  //initial value and number of
                           iterations
```

e. Then to generate all remaining key bytes, pth byte of the temporary array is iterated by p-1th byte of the actual key array.

```
key arr(p)=tent(k(p) , key arr(p-1))
```

F. Hénon map:

The value of $\alpha=1.4$ and $\beta=0.3$ are used as system parameters for this map. The starting value (X_0,Y_0) = (0.6314, 0.1894). Hénon map is a 2D map. the following method is used to generate a key array from it.

a. iterate the equation (3) 'r' times where 'r' is the number of rounds.
b. use the number upto 15 digits after the decimal point as integer to generate the key byte.

```
Zx = abs(Zx { integer part); // decimal part of x
Zy = abs(Zy { integer part); // decimal part of y
x new=mod(Zx*(10^15),256);
y new=mod(Zy*(10^15),256);
```

c. key byte is alternately taken as x new and y new.
d. use the decimal part Zx and Zy as starting values to generate the next key byte.
e. the number of iterations is changed to the absolute difference between x new and y new.

```
iterations=abs(y new - x new);
```

G. Gauss Iterated map:

The value of $\alpha = 6.2$ and $\beta = -0.48$ are used in equation (4). The starting value used is $X_0 = 0.8$. A approach similar to the one used in tent map for key generation is followed here.

a. X_0 is iterated 'r' (round) number of times.
b. The absolute value generated 'X_1' is mapped to [0-255] to produce the first byte of key K_1.
c. The subsequent key bytes K_n; $1 < n \leq 4r+8$; are generated by iterating X_{n-1}, K_{n-1} number of times.

H. Tinkerbell map:

The value of $\alpha = 0.9$, $\beta = -0.6013$, $\gamma = 2.0$, $\delta = 0.50$ are used as system variables in equation (5). The initial parameter used is (X0,Y0) = (-0.72, -0.64). Key generation function is as follows

a. initial parameter is iterated 'r' (round) number of times using equation (5).
b. The values of X and Y are mapped to [0-255] to generate X_{new}, Y_{new}.
c. The current key byte and the next iteration is derived by performing xor of current values of X_{new} and Y_{new}.

IV. EXPERIMENTATION RESULTS

The encryption/decryption of the `lena' image using different chaotic maps is shown in Fig. 2 to Fig. 5. We provide the results of statistical and differential analysis tests performed in the following subsections.

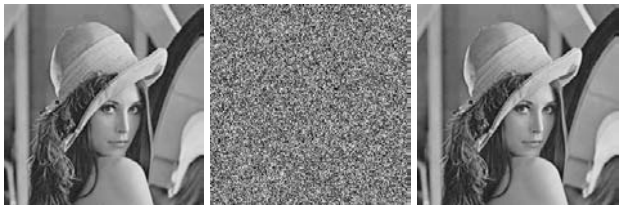

Figure 2. Encryption/Decryption of 'lena' plain image using Tent map.

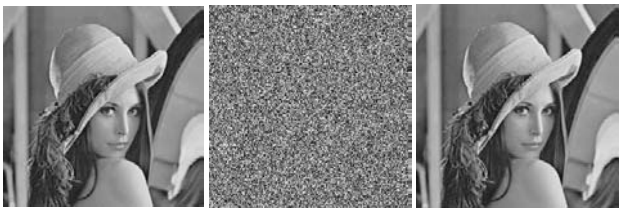

Figure 3. Encryption/Decryption of 'lena' plain image using Hénon map

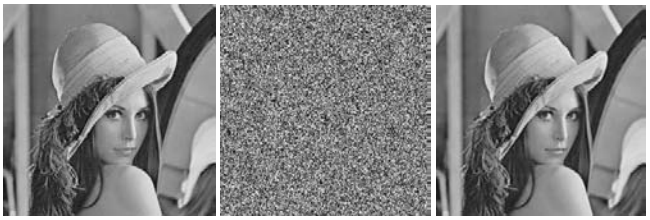

Figure 4. Encryption/Decryption of 'lena' plain image using Gauss iterated map

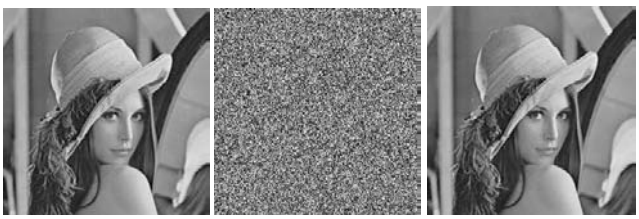

Figure 5. Encryption/Decryption of 'lena' plain image using Tinkerbell map

I. Statistical Analysis:

Statistical analysis of the cipher image is very important. We have shown histogram and information entropy analysis in this section.

a. Histogram Analysis:

For information to be secure from leakage, the encrypted image should bear very little resemblance to the plain image. Histograms plot number of pixels at each intensity level. This shows how pixels are distributed.

Fig. 6 depicts the histogram of the plain image 'lena'. Fig. 7(a-d) shows the histograms of the 'lena' image after encryption using different maps. They depict that the encryption does not leave any concentration of a single pixel value. All the maps show a very uniform distribution of pixel values.

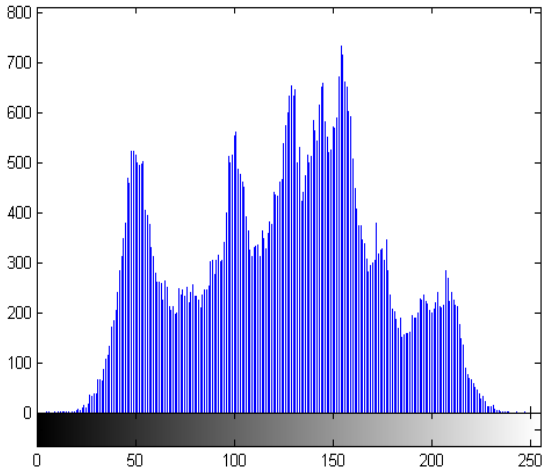

Figure 6. histogram of 'lena' plain image.

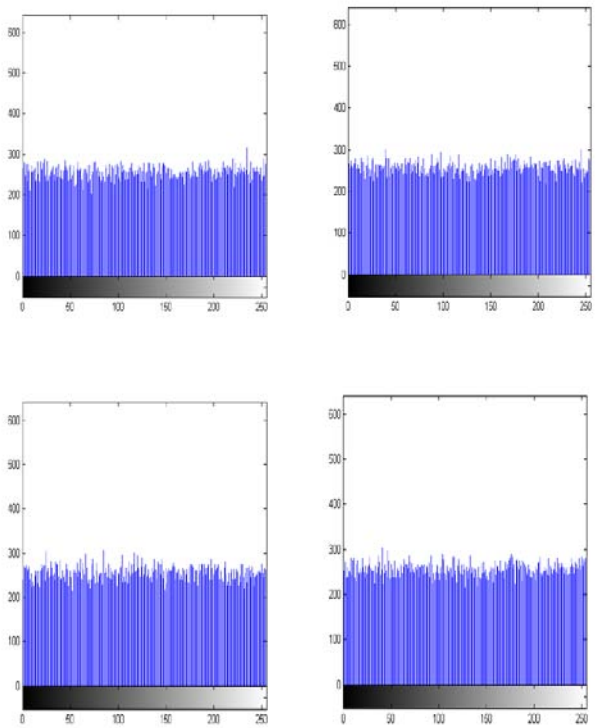

Figure 7. Histograms of encrypted 'lena' image shown in Figures 2-5 using (a) Tent map (b) Hénon map (c) Gauss iterated map (d) Tinkerbell Map (clockwise from top left)

b. Entropy Analysis:

Entropy is a measure of unpredictability of information content. Entropy of a source S is calculated as

$$H(s) = -\sum_{i=1}^{n} P_i \log_2(P_i) \qquad (6)$$

Where Pi means probability of occurrence of output units $a_1, a_2, \ldots, a_m$ from the information source. The ideal entropy of a random output should be 8. We have calculated the entropy of the 6 different cipher images of plain images shown in fig. 1. The results of the same have been provided in Table I.

Table I. entropy values obtained using different maps for 6 different images

Plaintext Images	*Chaotic Maps*			
	Tent Map	*Hénon Map*	*Gauss Iterated Map*	*Tinkerbell Map*
lena	7.9971	7.9975	7.9975	7.9969
cameraman	7.9976	7.9968	7.9974	7.9971
mandrill	7.9972	7.9973	7.9975	7.9973
boats	7.9974	7.9970	7.9971	7.9969
picard	7.9974	7.9973	7.9969	7.9973
lonesome	7.9969	7.9973	7.9971	7.9971

J. Differential Analysis:

To perform differential analysis, an adversary makes a small change (like changing a pixel value of an image) in the plaintext and tries to find out meaningful relationship between plaintext and ciphertext, by comparing the 2 different ciphertext of similar plaintext.

Two common measures used for differential analysis are: NPCR (net pixel change rate) and UACI (unified average changing intensity). NPCR shows the percentage rates at which pixels change in the cipher image when pixels of plain image are changed. UACI measures average intensity of difference between plain image and cipher image.

Let us consider 2 cipher images X_1 and X_2, obtained by plain images I_1 and I_2, where I_1 and I_2 have a single pixel difference. The pixel values at the grid position (i,j) for the two cipher images are denoted as $X_1(i,j)$ and $X_2(i,j)$. A bipolar array B is defined as follows

$$B(i,j) = \begin{cases} 0, & if\ X_1(i,j) = X_2(i,j) \\ 1, & if\ X_1(i,j) \neq X_2(i,j) \end{cases} \quad (7)$$

NPCR and UACI values are calculated as given in equation (8) and (9), where W and H denote width and height of the cipher images, T denotes the largest supported pixel value in the cipher images (255 in our case) and abs() computes the absolute value. The results of our analysis are provided in Table II.

$$\text{NPCR} = \frac{\sum_{i,j} B(i,j)}{W \times H} \times 100\% \quad (8)$$

$$\text{UACI} = \frac{1}{W \times H}\left[\sum_{i,j} \frac{abs(x_1(i,j) - x_2(i,j))}{T}\right] \times 100\% \quad (9)$$

V. CONCLUSION

Based on the results provided in section 4, following observations are made. The results are based on analysis performed on 6 different images (Fig. 1) and algorithm proposed in [3] keeping all other parameters constant. The histograms show almost similar distribution for different maps. Tent map shows the best average entropy values. The comparison based on NPCR and UACI values shows better performance of Gauss Iterated map.

VI. REFERENCES

[1] Menezes AJ, Oorschot PCV, and Vanstone SA. Handbook of applied cryptography. CRC Press: Florida, (1997).

[2] Schneier B. Applied cryptography: protocols algorithms and source code in C. Wiley: New York, (1996).

[3] Amin M, Faragallah OS, and Abd El-Latif AA. A chaotic block cipher algorithm for image cryptosystems. Commun Nonlinear Sci Numer Simulat, 15, 3484-3497 (2010).

[4] Chen G, Mao Y, and Chui CK. A symmetric image encryption based on 3D chaotic cat maps. Chaos Solitons and Fractals, 21, 749-61 (2004).

[5] Gao H, Zhang Y, Liag S, and Li D. A new chaotic algorithm for image encryption. Choas Solitons and Fractals, 29, 393-399 (2006)

[6] Guan Z-H, Huang F, and Guan W. Chaos based image encryption algorithm. Phys Lett A, 346, 153-7 (2005)

[7] Mao Y, Chen G, and Lian S. A novel fast image encryption scheme based on 3D chaotic Baker maps. Int J Bifurc Chaos, 14(10), 3613-24 (2004)

[8] Pareek NK, Patidar V, and Sud KK. Image encryption using chaotic logistic map. Image Vision Comput, 24, 926-34 (2006)

[9] Wong K-W, Kwok BS-H, and Law W-S. A fast image encryption scheme based on chaotic standard map. Phys Lett A, 372, 2645-52 (2008)

[10] Zhang L, Liao X, and Wang X. An image encryption approach based on chaotic maps. Chaos Solitons and Fractals, 24, 759-65 (2005)

Table II. NPCR and UACI values obtained using different maps for encryption of 6 plain images and same images with 1 pixel changed.

	Plain Images	***Tent Map***	***Hénon Map***	***Gauss Iterated Map***	***Tinkerbell Map***
NPCR	lena	99.5834	99.6536	99.5987	99.5941
	cameraman	99.6521	99.5895	99.6231	99.6231
	mandrill	99.5941	99.5956	99.6262	99.5651
	boats	99.6292	99.5819	99.6002	99.6094
	picard	99.5941	99.6277	99.6078	99.6445
	lonesome	99.5895	99.6262	99.6231	99.6048
UACI	lena	33.4988	33.4922	33.3757	33.3807
	cameraman	33.4608	33.4343	33.6171	33.5629
	mandrill	33.4482	33.4534	33.5030	33.3791
	boats	33.3665	33.5276	33.5602	33.3441
	picard	33.4281	33.4294	33.6319	33.3522
	lonesome	33.4151	33.5061	33.4192	33.4730