A Simple Method for Image Encryption Using Chaotic Logistic Map

Kamal Jadidy Aval, Morteza Sabery Kamarposhty, Masumeh Damrudi

Islamic Azad university of Firuzkooh,
Corresponding addresses
{k.jadidy, msaberyk}@gmail.com, info@damrudi.ir

Abstract: In this paper, a new method is suggested for image hiding using chaos signals. The combination of movement of pixels and adjusting quantity of gray level is simultaneously used in this method. The process of moving pixels is done by an order which is taken from logistic map and for adjusting gray level the order of standing of bits of pixel's quantity of gray level is changed by the means of chaos signal as well. Experimental results show that this method has a proper efficiency for prevalent terms. For example the amount of entropy obtained by this method is around 7.9949 that is so near to 8 its ideal amount.

Keywords: Image Encryption, Chaos, Logistic Map, Brute-Force Attacks.

1. Introduction

Increasing growth of multimedia production and sharing the digital information, caused to protection of private information have more importance at the networks.There are many suggested algorithms to do this. [1, 2, 3, 4] Recently, along with the rapid development of theory and application of chaos, many researchers are now focusing on the chaotic cryptography. A lot of image encryption schemes based on chaos theory have been presented [5, 6, 7, 8, 9, 10] These methods have been motivated by the chaotic properties such as ergodicity and sensitive dependence on initial conditions and system parameters, in addition to complex dynamics and deterministic behaviors.

A method based on shuffling the positions of the pixels of the plain-image is proposed at [5]. A new chaotic key based image encryption algorithm (CKBDA) is proposed at [7] which a chaotic signal use to change gray values of the images but this method haven't suitable security so researchers try to use shuffling the positions and changing the gray values of the pixels simultaneously.[8, 9, 10, 11]

For instance, in [8, 9, 10, 11] the two-dimensional chaotic map is generalized to 3D for designing a real-time secure symmetric encryption scheme. The new scheme employs the 3D map to shuffle the positions of image pixels and uses another chaotic map to confuse the relationship between the cipher-image and plain-image.

In [12], A fast chaos-based image encryption system with stream cipher structure is proposed. In order to achieve a fast throughput and facilitate hardware realization, 32-bit precision representation with fixed point arithmetic is assumed. The major core of the encryption system is a pseudo-random keystream generator based on a cascade of chaotic maps, serving the purpose of sequence generation and random mixing. Unlike the other existing chaos-based pseudo-random number generators, the proposed keystream generator not only achieves a very fast throughput, but also passes the statistical tests of up-to-date test suite even under quantization.

In [14], eight different types of operations are used to encrypt the pixels of an image and which one of them will be used for a particular pixel is decided by the outcome of the logistic map. To make the cipher more robust against any attack, the secret key is modified after encrypting each block of sixteen pixels of the image.

In this paper we introduce a algorithm based on shuffling the positions and changing the gray values of the pixels and evaluate its resistance via different attacks.

2. The Method

In this section we first introduce the applied chaotic system and then introduce the proposed approach.

2.1 Chaotic System

A chaotic system has a noise like behaviour while is exactly deterministic so if we have its parameters and initial values, we can reproduce it. These signals are extremely sensitive to initial conditions. one of the most famous chaotic system is logistic map, that is a nonlinear return map given by

$$X_{n+1} = rX_n(1 - X_n) \qquad (1)$$

This map creates chaotic orbits for $r > 3.83$. Figure 1 shows the plot of x(n) vs n for $X_0 = 0.5$ and r =3.9999 after 500 iteration.

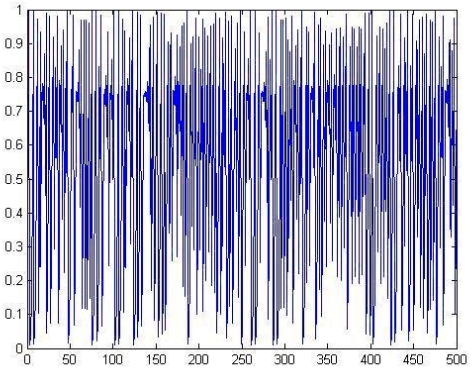


Figure 1. Chaotic behaviour of logistic map with $X_0 = 0.5$ and r=3.9999

2.2 The Algorithm

The applied algorithm is explained step by step:

Step1: The method use a key to improve the security of the method that is a key with 80 bit defined as

$$K = K_0, K_1, ..., K_{19}(HexaDecimal) \quad (2)$$

That K_i can be one of alphanumeric characters (0-9 and A-F), or can be defined using ASCII codes as

$$K = K_0, K_1, ..., K_9(Ascii) \quad (3)$$

In this key K_i, defines a 8 bit block of the key.

Step2: To use logistic map, we need to define an initial value X_0, to do this , we first convert the key to a binary form

$$K = \begin{bmatrix} K_{01}, K_{02}, K_{03}, K_{04}, K_{05}, K_{06}, K_{07} \\ \vdots \\ K_{90}, K_{91}, K_{92}, K_{93}, K_{94}, K_{95}, K_{96}, K_{97} \end{bmatrix} \quad (4)$$

That K_{ij} is the jth bit of the ith part of the key. The X_{01} value calculates as

$$X_{01} = \begin{bmatrix} K_{01} \times 2^{39} + K_{03} \times 2^{38} + K_{05} \times 2^{37} + K_{07} \times 2^{36} \\ \vdots \\ + K_{91} \times 2^3 + K_{93} \times 2^2 + K_{95} \times 2^1 + K_{97} \times 2^0 \end{bmatrix} \quad (5)$$

Which n = 0, 1, ..., 9 defines the key number and K_{ij} defines the jth bit from the ith part of the key so that j = 1, 3, 5, 7. Also X_{02} can be calculated as follow

$$X_{02} = \begin{bmatrix} K_{00} \times 2^{39} + K_{02} \times 2^{38} + K_{04} \times 2^{37} + K_{06} \times 2^{36} \\ \vdots \\ + K_{90} \times 2^3 + K_{92} \times 2^2 + K_{94} \times 2^1 + K_{96} \times 2^0 \end{bmatrix} \quad (6)$$

Which n = 0, 1, ..., 9 defines the key number and K_{ij} defines the jth bit from the ith part of the key so that j = 0, 2, 4, 6. Finally the initial value X_0 can be calculated as

$$X_0 = (X_{01} + X_{02}) \bmod 1 \quad (7)$$

Step3: Convert the image with $M \times M$ pixels to P blocks with $N \times N$ size. we have

$$P = (M/N)^2 \quad (8)$$

As can be seen in Figure 1, The map (1) defined in [0, 1]. We can divide this interval to P sections which its size is

$$\varepsilon = 1/P \quad (9)$$

and the ith interval is

$$((i-1)\varepsilon, i\varepsilon) \quad (10)$$

Then X_i will be calculated using (1). After defining the number of this interval as the first order, provided that the orbit don't be lie at this interval, this will repeat until the orbit be lie at all P intervals. At least we have a order like

$$Iteration = (it_1, it_2, ..., it_r) \quad (11)$$

Which $r \le P$. This order defines the location of the image pixels. In this way the interval of the ith block will be

$$\begin{aligned} x_{start} &= \left(Round((it_i - 1)/\sqrt{P})\right) \times N \\ y_{start} &= \left((it_i - 1) \bmod \sqrt{P})\right) \times N \\ x_{end} &= x_{start} + N \\ y_{end} &= y_{start} + N \end{aligned} \quad (12)$$

Step4: We set the gray values of the main image at a one dimensional matrix so that values of rows make the matrix as

$$V = \{v_1, v_2, ..., v_{M \times M}\} \quad (13)$$

So the number of the block that we need to ith pixel replace to, is

$$\begin{aligned} block\_no &= ((i-1) \bmod P) + 1 \\ iter &= iteration(block\_no) \end{aligned} \quad (14)$$

Initial point of this block, obtain from

$$\begin{aligned} x_{start} &= \left(Round((iter - 1)/\sqrt{P})\right) \times N \\ y_{start} &= \left((iter - 1) \bmod \sqrt{P})\right) \times N \end{aligned} \quad (15)$$

And the variations of the inside of the block is

$$\begin{aligned} dif &= (i-1) divP \\ x_{dif_i} &= ((dif - 1) divN) + 1 \\ y_{dif_i} &= ((dif - 1) \bmod N) + 1 \end{aligned} \quad (16)$$

So the position of ith pixel will be

$$\begin{aligned} x_{pos} &= x_{start} + x_{dif} \\ y_{pos} &= y_{start} + y_{dif} \end{aligned} \quad (17)$$

Step5: We also use signal (1) to change every pixel's gray level as the structure below:

First divide the interval [0, 1] to 8 parts, Then we make a matrix with one line and eight column named Map that it's rudimentary amount is zero. After that we figure the first quantity of this signal X_1 by the rudimentary amount X_0. suppose that this amount X_1 is located in i th span . in that way we'll have:

$$Map(1,i)=1 \quad (18)$$

Then we make second amount of signal X_2. Now if this amount is located in j th span,in this circumstances if the Map(1, j) has not been numbered before, we'll have:

$$Map(1,j)=2 \quad (19)$$

and this procedure will continue until all the elements of matrix are numbered.finally we'll have a matrix named Map like below:

$$Map=\{m_1,m_2,.....,m_8\} \quad (20)$$
$$k=1,2,3,4,5,6,7,8 \qquad 1\le m_k \le 8$$

Step6: We convert the quantity of gray level of supposed pixel into binary:

$$B=b_8b_7.....b_1 \quad (21)$$

that we'll have $b_i \in [0,1]$

we'll define the converted quantity of gray level like this:

$$C=c_8c_7.....c_1 \quad (22)$$

that the amount of c_i will be equal with the m_i th bit of B. For instance if Map matrix has been calculated like the following form:

$$Map=\{5,4,2,7,3,1,8,6\} \quad (23)$$

so in this circumstances C will be:

$$C=b_5b_4b_2b_7b_3b_1b_8b_6 \quad (24)$$

These procedures will be done consecutively for each pixel so that the last produced amount for creating a Map matrix for current pixel will be used as the chaos signal's rudimentary quantity for making Map matrix for next pixel.

3. Experimental Results

A good encryption method, must be stable via all attack methods like cryptanalytic, brute-force and statistical attacks. In this section we examine the proposed method through statistical analysis, sensitivity to key changes and key space analysis.

3.1 Statistical Analysis

To examine the stability via statistical attacks,we calculat the histogram and correlation between adjacent pixels, fo many common images.

3.1.1 Histogram Analysis

Histogram shows the number of pixels for any gray value in the image. Figures 2(*a*)*and*(*e*) shows the plain image and encrypted image and their histogram of three channels for two images are shown in Figures 2(*b*), (*c*), (*d*), (*f*), (*g*) *and* (*h*).

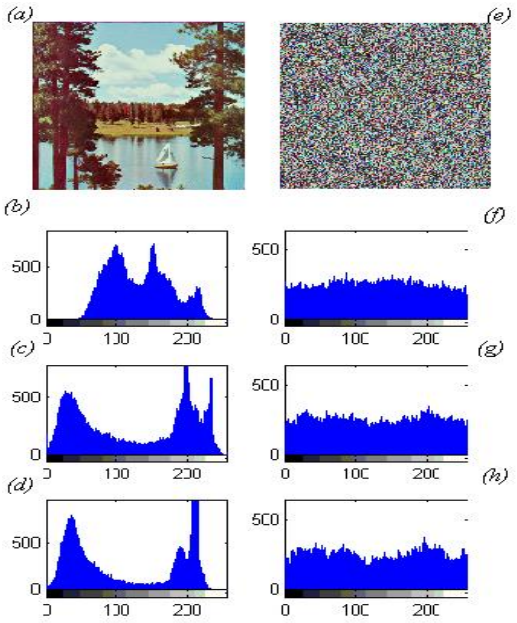
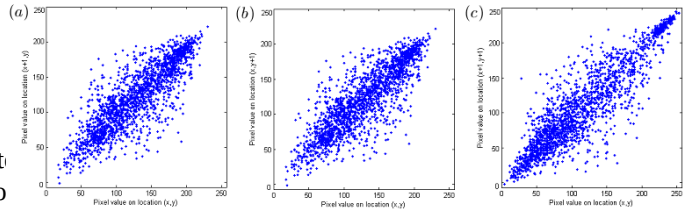


Figure 2. Histogram analysis: Frame (a) shows a plain image. Frames (b), (c) and (d) respectively, show the histograms of red, green and blue channels of the plain image shown in figure (a). Frame (e) show the encrypted image of the plain image shown in frame (a) using the secret key 'ABCDEF0123456789ABCD' (in hexadecimal). Frames (f), (g) and (h) respectively show the histograms of red, green and blue channels of encrypted image shown in figure (e).

These figures show the encrypted image with the key *"ABCDEF*0123456789*ABCD"* has a uniform histogram and so then method is resistive to statistical attacks.

3.1.2 Correlation coefficient Analysis

In this section the horizontal, vertical and diagonal correlation coefficient of the pixels studied. To do this we choose 2048 pairs of horizontal, vertical and diagonal adjacent pixels randomly. Figure 3 show the distribution of two horizontally, vertically and diagonally adjacent pixels in plain image and encrypted image.



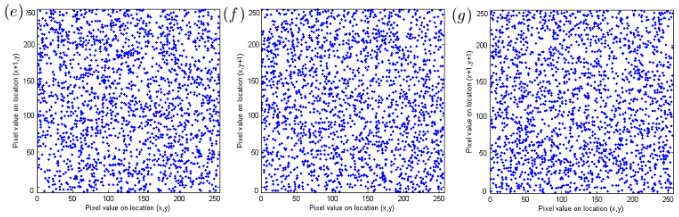


Figure 3. Correlation of two adjacent pixels: Frame (a), (b) and (c) respectively, show the distribution of two horizontally,vertically and diagonally adjacent pixels in plain image shown in figure 2(a) . Frame (d), (e) and (f) respectively, show the distribution of two horizontally, vertically and diagonally adjacent pixels in encrypted image shown in figure 2(b).

The correlation coefficient of two adjacent pixels calculated using

$$E(x)=\frac{1}{N}\sum_{i=1}^{N}x_i$$
$$D(x)=\frac{1}{N}\sum_{i=1}^{N}\left(x_i-E(x_i)\right)^2$$
$$\operatorname{cov}(x,y)=\frac{1}{N}\sum_{i=1}^{N}\left(x_i-E(x_i)\right)\left(y_i-E(y_i)\right) \qquad (23)$$
$$r_{xy}=\frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

In Table 1, We have given the correlation coefficient for the plain and encrypted images shown in figure 2(a) and 2(b). It is clear from the Table 1 that there is negligible correlation between the two adjacent pixels in the encrypted image. However, the two adjacent pixels in the plain image are highly correlated.

Table 1. Correlation coefficient of two adjacent pixels in plain image and encrypted image

	Plain image	Encrypted image
Horizontal	0.9394	-0.0274
Vertical	0.9205	-0.0092
Diagonal	0.9129	0.0034

NPCR and UACI are two criterion that examine the effect of changing just one pixel in the plain image on the encrypted image [13]. NPCR is the pixel change rate at the encrypted image for changing a pixel at the plain image and UACI is the mean of these changes. NPCR and UACI are given by

$$NPCR=\frac{\sum_{i,j}D(i,j)}{W\times H}\times 100\% \qquad (24)$$
$$UACI=\frac{1}{W\times H}\left[\sum_{i,j}\frac{|C_1(i,j)-C_2(i,j)|}{255}\right]\times 100\%$$

at which H and W are the height and width of the image and $C1$ and $C2$ are two encrypted images obtained from two plain image with just one different pixel And D is

$$D(i,j)=\begin{cases}1 & \text{if } C_1(i,j)=C_2(i,j)\\ 0 & \text{otherwise}\end{cases} \qquad (25)$$

The results of an 256 × 256 image are NPCR=0.431 % UACI=0.334 % that shows the method is resistive to differential attacks.

3.1.3 Key Sensitivity Analysis

A good encryption algorithm, must be sensitive to the small variations of the key. Changing a bit must give a different result from the plain image.

Figure 4(*a*) *and* (*b*) shows the plain image and encrypted images with the key 'ABCDEF0123456789ABCD' and Figure 4(*c*)*and*(*d*) shows the encrypted images with keys 'BBCDEF0123456789ABCD' and 'ABCDEF0123456789ABCE'.

To compare the results, the correlation coefficient between selected point for any pairs of encrypted image calculated. Table 3 shows this results. The results show that the method is sensitive to small variations of the key.

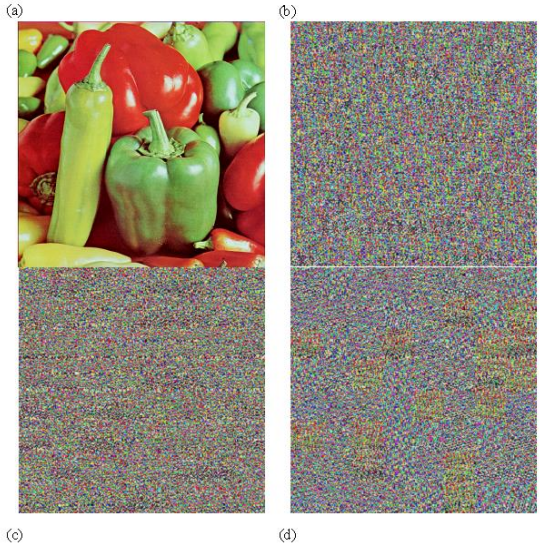


Figure 4. Sensitivity to the small variations of the key. Frame (a) and (b) shows the plain image and encrypted images with the key 'ABCDEF0123456789ABCD' and Frame (c) and (d) shows the encrypted images with keys 'BBCDEF0123456789ABCD' and 'ABCDEF0123456789ABCE'.

Table 3. Correlation coefficient between selected point for any pairs of encrypted image with 1 bit difference in secret key.

Image 1	Image 2	Correlation coefficient
*Fig*5(*b*)	*Fig*5(*c*)	−0.0115
*Fig*5(*c*)	*Fig*5(*d*)	0.0008
*Fig*5(*b*)	*Fig*5(*d*)	−0.0087

Moreover, in Fig 5, we have shown the result of some attempts to decrypt an encrypted image with slightly different secret keys than the one used for the encryption of the original image. Particularly, in frames (a) and (b) respectively, the original image and encrypted image produced using the secret keys 'DBACEF0123456789FDCD'(in hexadecimal) are shown whereas in frame (c) and (d) respectively, the images after the decryption of the encrypted image(shown in frame (b)) with the secret key 'DBACEF0123466789FDCD'(in hexadecimal) and 'DBACFF0123456789FDCD'(in hexadecimal). It is clear that the decryption with a slightly different key fails completely and hence the proposed image encryption procedure is highly key sensitive.

Figure 5. Key sensitivity test: Frame (a) and (b) respectively, show a plain image and its encrypted image using the secret key DBACEF0123456789FDCD'(in hexadecimal). Frames (c) and (d) respectively, show the image after the decryption of the encrypted image shown in frame (b) using the secret key 'DBACEF0123466789FDCD'(in hexadecimal) and 'DBACFF0123456789FDCD'(in hexadecimal).

3.1.4 Key Space Analysis

The key space must be sufficiently large to the method be resistive via brute-force attacks. The keys of the proposed method must be choose from $2^{80}\left(\approx 1.20893\times 10^{24}\right)$ keys that shows the method is resistive via brute-force attacks.

3.1.53.1.5 Information Entropy

Information entropy is a common criterion that shows the randomness of the data. The mathematical theory of information entropy introduced by Claude E. Shanon at 1949 [15]. One of the most famous formulas of the information entropy is

$$H(S)=\sum_{i=0}^{2N-1}P(s_i)\log\left(\frac{1}{P(s_i)}\right) \qquad (26)$$

That N is the number of gray level in the image (256 for 8 bit images) and $P(s_i)$ shows the probability ith gray level. For a ideal random image, the value of information entropy is 8. The predictability of the method decreases when the information entropy tends to 8. For the proposed method we obtain information entropy= 7.9949 that is very close to the ideal value.

4. Conclusion

In this paper, a new method is proposed for image encryption based on a chaotic maps. Replacing the image pixels and changing the gray level values used simultaneously. The experimental results show that the proposed method based on chaotic shuffling and changing the gray value is resistive via different attacks like cryptanalytic, brute-force and statistical attacks.

References

[1] S.S. Maniccam, N.G. Bourbakis. Image and video encryption using SCAN patterns. *Pattern Recognition* 2004(37): 725-737.

[2] C.-C. Chang, M.-S. Hwang, T.-S. Chen. A new encryption algorithm for image cryptosystems. *Journal of Systems and Software* 2001(58)(2): 83-91.

[3] YN. Bourbakis, C. Alexopoulos. Picture data encryption using scan patterns, *Pattern Recognition* 1992(25)(6):567-581.

[4] H. Cheng, X.B. Li. Partial encryption of compressed image and videos, *IEEE Transaction of Signal Processing* 2000 (48)(8): 2439-2451.

[5] J. Fridrich. Symmetric ciphers based on two dimensional chaotic maps. *International Journal of Bifurcation and Chaos* 1998 (8)(6): 1259-1284.

[6] J.-C. Yen, J.-I. Guo. A New Chaotic Key-Based Design for Image Encryption and Decryption. *Proceedings IEEE International Conference on Circuits and Systems* 2000 948-952.

[7] S. Li, X. Zheng. Cryptanalysis of a Chaotic Image Encryption Method, *Proceedings IEEE International Symposium on Circuits and Systems USA* 2002 708-711.

[8] F. Beldhouche, U. Qidwai. Binary Image Encoding Using 1D Chaotic Maps. *IEEE Annual Technical Conference* 2003 39-43.

[9] Y. Mao, G. Chen. Chaos-Based Image Encryption, Handbook of Computational Geometry for Pattern

Recognition. Computer Vision, Neurocomputing and Robotics, *Springer-Verlag, Berlin*, 2003.

[10] Chen GR, Mao YB, et al. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons & Fractals* 2004 (21):749-761.

[11] Gao T, Chen Z. Image encryption based on a new total shuffling algorithm . *Chaos, Solitons & Fractals* (2007); doi:10.1016/j.chaos.2006.11.009

[12] H.S. Kwok, Wallace K.S. Tang. A fast image encryption system based on chaotic maps with finite precision representation. *Chaos, Solitons & Fractals* 2007(32): 1518-1529.

[13] H. Gao, Y. Zhang, S. Liang, D. Li. A new chaotic algorithm for image encryption. Chaos,*Solitons & Fractals* 2006 (29):393-399.

[14] N.K. Pareek, Vinod Patidar , K.K. Sud, Image encryption using chaotic logistic map, *Image and Vision Computing* 2006 (24) 926–934

[15] C. E. Shannon. Communication theory of security systems. *The Bell Syst Tech J* 1949 (28) 656-715.