An Image Encryption Scheme Based on Cat Map and Hyperchaotic Lorenz System

Jian Zhang
Shenyang Fire Research Institute, Shenyang 110034, China
E-mail: zhangjianfire@163.com

***Abstract*—In recent years, chaos-based image cipher has been widely studied and a growing number of schemes based on permutation-diffusion architecture have been proposed. However, recent studies have indicated that those approaches based on low-dimensional chaotic maps/systems have the drawbacks of small key space and weak security. In this paper, a security improved image cipher which utilizes cat map and hyperchaotic Lorenz system is reported. Compared with ordinary chaotic systems, hyperchaotic systems have more complex dynamical behaviors and number of system variables, which demonstrate a greater potential for constructing a secure cryptosystem. In diffusion stage, a plaintext related key stream generation strategy is introduced, which further improves the security against known/chosen-plaintext attack. Extensive security analysis has been performed on the proposed scheme, including the most important ones like key space analysis, key sensitivity analysis and various statistical analyses, which has demonstrated the satisfactory security of the proposed scheme.**

Keywords- image cipher; permutation-diffusion; cat map; hyperchaotic Lorenz system

I. INTRODUCTION

Chaos-based image cipher has been widely investigated over the last decade or so to meet the increasing demand for online secure image transmission over open networks. Image encryption is different from text encryption due to some intrinsic features of images such as bulk data capacity and high redundancy, which are generally difficult to be handled by using conventional block ciphers. With the desirable properties of pseudo-randomness, ergodicity, high sensitivity to initial conditions/parameters, chaotic maps/ systems have demonstrated great potential for information especially multimedia encryption. After Fridrich proposed the first chaotic image encryption scheme in 1998 [1], increasing researches of image encryption are based on chaotic systems [2-11]. Chaos-based algorithms have shown their superior performance in the aspect of complexity, speed, computing power and security.

Though there have been many publications of chaos-based image cipher, only a few high-dimensional chaotic systems are integrated into these cryptosystems. Recent studies have pointed out that image cryptosystems built upon low-dimensional chaotic maps/systems have the advantages of high-level efficiency and simplicity, but their weaknesses, such as small key space and weak security, are also obvious [12-14]. To overcome these drawbacks, this paper proposes a security improved image cipher using cat map and hyperchaotic Lorenz system. Compared with ordinary chaotic systems, hyperchaotic systems, possessing more than one positive Lyapunov exponents, have more complex dynamical behaviors and number of system variables [15-16]. This implies that the cryptosystems based upon hyperchaotic system have stronger unpredictability and larger key space. Moreover, a plaintext related key stream generation strategy is introduced in the diffusion stage, which further improves the security against known/chosen-plaintext attack. Thorough experimental tests are carried out with detailed analysis, demonstrating the high security of the proposed scheme. The remainder of this paper is organized as follows. Section 2 discusses the image shuffling algorithm using cat map. Then the hyperchaotic Lorenz system based image diffusion process is described in Section 3. In Section 4, we analyze the security of the proposed image cipher, and the conclusions are drawn in the last section.

II. IMAGE PERMUTATION USING CAT MAP

Image data have strong correlations among adjacent pixels. Statistical analysis on large amounts of images indicates that averagely adjacent 8 to 16 pixels are correlative in horizontal, vertical, and also diagonal directions for both natural and computer-graphical images. To erase such correlations, the Arnold cat map is employed to shuffle the pixel positions of the plain-image.

The discretized cat map is a chaotic bijection of the lattice $N\times N$ onto itself. It is defined by

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod N, \tag{1}$$

where p and q are control parameters, and mod(x, N) divides x by N and returns the remainder of the division. The map is invertible and area-preserving as the matrix has determinant 1.

The inverse transform of the discretized cat map for de-shuffling is easily found to be given by

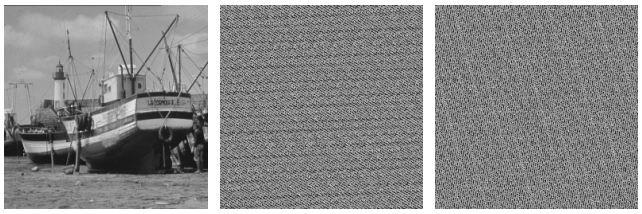
$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} pq+1 & -p \\ -q & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod N. \tag{2}$$

Obviously, the two parameters p and q can serve as the permutation key and their values can be any positive integer. However, as the four-tuple [1, (p+k_1N), (q+k_2N),

978-1-4799-6023-1/15 $31.00 © 2015 IEEE
DOI 10.1109/CICT.2015.134

$(p+k_1N)(q+k_2N)+1]$ generates the same cipher as the four-tuple $[1, p, q, (pq+1)]$ for any $k_1, k_2, k_3, k_4 \in Z$, the values of p, q should be restricted to the set N. Therefore, the total number of ciphering keys the map can provide is $(N^2)^m$, where m is the iteration times.

The application of the cat map to a grayscale test image with 256×256 size is demonstrated in Fig. 1. Fig. 1(a) shows the plain-image, and Figs. 1(b)-(c) show the results of applying the discretized cat map once and two times, respectively. The ciphering key is (p=208, q=231).



(a) (b) (c)

Figure 1. The application of the cat map. (a) The test image 256×256 pixels with 256 gray levels. (b) The test image after applying the cat map once. (c) The test image after applying the cat map two times.

As can be seen from Fig. 1, after only one round permutation, the correlation among the adjacent pixels is effectively erased and the image is unrecognizable. Unfortunately, the statistical property of the shuffled image is identical to that of the plain-image as the operation only shuffles the pixels positions without changing their values. Therefore, the shuffled image is weak against statistical analysis and known/chosen-plaintext attack. As a remedy, a diffusion procedure is introduced next to improve the security.

III. Image Diffusion Using Hyperchaotic Lorenz System

In the diffusion stage, the pixel values are modified sequentially to confuse the relationship between cipher-image and plain-image. In the present paper, a new hyperchaotic system [16], which is obtained by adding a nonlinear quadratic controller to the second equation of the Lorenz chaotic system, is employed to generate the diffusion key stream. The system is described by

$$\begin{cases} \dot{x} = a(y-x), \\ \dot{y} = cx + y - xz - w, \\ \dot{z} = xy - bz, \\ \dot{w} = kyz, \end{cases} \tag{3}$$

where a, b, c are the system parameters, and k is the control parameter, determining the chaotic attractor and bifurcation of the system. When parameters a=10, b=8/3, c=28 and $0 < k < 0.152$, the system is hyperchaotic. The initial system variables x_0, y_0, z_0 and w_0 are used as the diffusion key.

For most existing permutation-diffusion type image ciphers, the key stream quantified from the system variables only depends on the key (the initial values/parameters of the chaotic system). The same key stream is used to encrypt different plain-images if the key remains unchanged. Therefore, an opponent may obtain the key stream by known or chosen-plaintext attacks, i.e., by encrypting some special plaintext sequences and then comparing them with the corresponding ciphertext sequences. In the present scheme, the key stream elements quantified from each iteration of the hyperchaotic system are reordered according to the previous plain-pixel. As a result, the quantified key stream is related not only to the key but also to the plain-image, which effectively addresses the aforementioned flaw. The detailed diffusion process is described as follows:

Step 1: The pixels of the shuffled image are arranged to a vector $p=\{p_1, p_2, \ldots, p_{N\times N}\}$ in the order from left to right, top to bottom.

Step 2: Iterate Eq. (3) for N_0 times to avoid the harmful effect of transitional procedure, where N_0 is a constant. To solve the equation, fourth-order Runge-Kutta method is employed, as given by

$$\begin{cases} x_{n+1} = x_n + (h/6)(K_1 + 2K_2 + 2K_3 + K_4), \\ y_{n+1} = y_n + (h/6)(L_1 + 2L_2 + 2L_3 + L_4), \\ z_{n+1} = z_n + (h/6)(M_1 + 2M_2 + 2M_3 + M_4), \\ w_{n+1} = w_n + (h/6)(N_1 + 2N_2 + 2N_3 + N_4), \end{cases} \tag{4}$$

where

$$\begin{cases} K_j = a(y_n - x_n) \\ L_j = cx_n + y_n - x_n z_n - w_n \\ M_j = x_n y_n - bz_n \\ N_j = ky_n z_n \end{cases} \quad (j=1),$$

$$\begin{cases} K_j = a[(y_n + hL_{j-1}/2) - (x_n + hK_{j-1}/2)] \\ L_j = c(x_n + hK_{j-1}/2) + (y_n + hL_{j-1}/2) \\ -(x_n + hK_{j-1}/2)(z_n + hM_{j-1}/2) - (w_n + hN_{j-1}/2) \\ M_j = (x_n + hK_{j-1}/2)(y_n + hL_{j-1}/2) \\ -b(z_n + hM_{j-1}/2) \\ N_j = k(y_n + hL_{j-1}/2)(z_n + hM_{j-1}/2) \end{cases}$$

$(j=2,3)$,

$$\begin{cases} K_j = a[(y_n + hL_{j-1}) - (x_n + hK_{j-1})] \\ L_j = c(x_n + hK_{j-1}) + (y_n + hL_{j-1}) \\ -(x_n + hK_{j-1})(z_n + hM_{j-1}) - (w_n + hN_{j-1}) \\ M_j = (x_n + hK_{j-1})(y_n + hL_{j-1}) - b(z_n + hM_{j-1}) \\ N_j = k(y_n + hL_{j-1})(z_n + hM_{j-1}) \end{cases} \quad (j=4),$$

and the step h is chosen as 0.0005.

Step 3: The hyperchaotic Lorenz system is iterated continuously. For each iteration, we can obtain four key stream elements from the current state of the chaotic system according to

$$k_{\varphi n} = \mathrm{mod}[round((abs(\varphi_n) - floor(abs(\varphi_n)))\times 10^{14}), L]$$

$$(\varphi \in \{x, y, z, w\}), \quad (5)$$

where *abs*(x) returns the absolute value of x, *floor*(x) returns the value of x to the nearest integers less than or equal to x, *round*(x) rounds x to the nearest integers, mod(x, y) returns the remainder after division, and L is the color level.

Step 4: Suppose Ω is the set of all possible arrangements of $k_{\varphi n}$. A certain permutation $k_{\varphi n}' = (k_{xn}', k_{yn}', k_{zn}', k_{wn}')$ is selected from Ω according to

$$k_{\varphi n}' = p' \% M + 1, \quad (6)$$

where p' is the previously operated plain-pixel and M=24 the total number of permutations of $k_{\varphi n}$. The initial value of p' can be set as a constant.

Step 5: Mask the plain-pixel according to Eq. (7).

$$\begin{cases} c_{4\times(n-1)+1} = k_{xn}' \oplus \{[p_{4\times(n-1)+1} + k_{xn}'] \bmod L\} \oplus c_{4\times(n-1)}, \\ c_{4\times(n-1)+2} = k_{yn}' \oplus \{[p_{4\times(n-1)+2} + k_{yn}'] \bmod L\} \oplus c_{4\times(n-1)+1}, \\ c_{4\times(n-1)+3} = k_{zn}' \oplus \{[p_{4\times(n-1)+3} + k_{zn}'] \bmod L\} \oplus c_{4\times(n-1)+2}, \\ c_{4\times(n-1)+4} = k_{wn}' \oplus \{[p_{4\times(n-1)+4} + k_{wn}'] \bmod L\} \oplus c_{4\times(n-1)+3}, \end{cases} \quad (7)$$

where n = 1, 2, . . . represents the nth iteration of the hyperchaotic system and $c_{4\times(n-1)+m}$ (m = 1, 2, 3, 4) are the output cipher-pixels, respectively, and $\oplus$ performs bit-wise exclusive OR operation. One may also set initial value c_0 as a constant.

Step 6: Return to *Step 3* until all the pixels in vector p are encrypted. Finally the encrypted pixel set c={c_1, c_2, …, $c_{N\times N}$} is rearranged to a $N\times N$ matrix and the cipher-image is obtained.

The decryption procedure is similar to that of the encryption process described above, and the inverse of Eq. (7) is given by

$$\begin{cases} p_{4\times(n-1)+1} = [k_{xn}' \oplus c_{4\times(n-1)+1} \oplus c_{4\times(n-1)} + L - k_{xn}'] \bmod L, \\ p_{4\times(n-1)+2} = [k_{yn}' \oplus c_{4\times(n-1)+2} \oplus c_{4\times(n-1)+1} + L - k_{yn}'] \bmod L, \\ p_{4\times(n-1)+2} = [k_{yn}' \oplus c_{4\times(n-1)+2} \oplus c_{4\times(n-1)+1} + L - k_{yn}'] \bmod L, \\ p_{4\times(n-1)+4} = [k_{wn}' \oplus c_{4\times(n-1)+4} \oplus c_{4\times(n-1)+3} + L - k_{wn}'] \bmod L. \end{cases} \quad (8)$$

IV. Security Analysis

In this section, some security analysis has been performed on the proposed scheme, including the most important ones like statistical analysis, key space analysis, and key sensitivity analysis, which has demonstrated the satisfactory security of the proposed scheme, as discussed in the following.

A. Statistical analysis

Due to the inherent presentation formats of images, statistical analysis provides an effective way to analyze an image cryptosystem and several statistical attacks have been devised on them. To prove the robustness of the proposed scheme in the aspect of statistical attacks, two tests are performed in this subsection, including analyses of histogram as well as correlation of adjacent pixels.

1) Histogram

The distribution of ciphertext is of much importance to a cryptosystem. It should hide the redundancy of plaintext and should not leak any information about the plaintext or the relationship between plaintext and ciphertext. The histograms of plain-image (Fig. 2(a)) and its corresponding cipher-image (Fig. 2(c)) produced by the proposed scheme are shown in Figs. 2(b), (d), respectively. It's clear from Fig. 2(d) that the histograms of the cipher-image are fairly uniform and significantly different from that of the plain-image and hence does not provide any clue to employ histogram analysis.

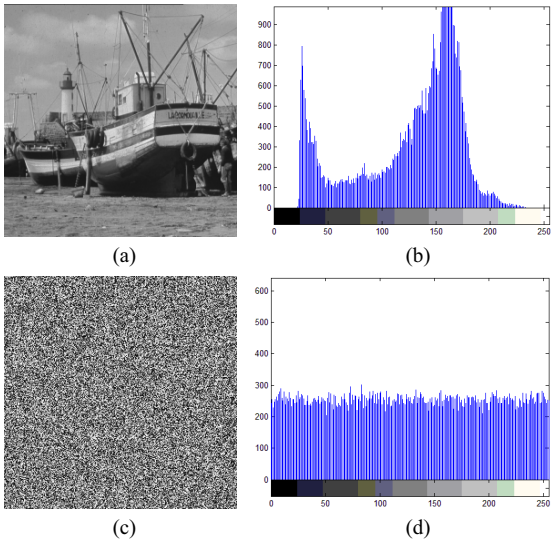
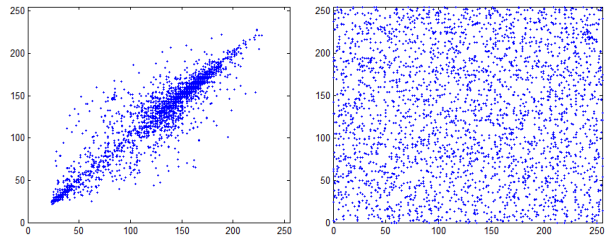


Figure 2. Histograms of plain-image and cipher-image. (a) plain-image. (b) histogram of plain-image. (c) cipher-image. (d) histogram of cipher-image.

2) Correlation of adjacent pixels

For any ordinary image having meaningful visual content, each pixel is highly correlated with its adjacent pixels either in horizontal, vertical or diagonal direction. However, an efficient image cryptosystem should procedure cipher-images with sufficiently low correlation in the adjacent pixels. The visual testing of the correlation of adjacent pixels can be done by plotting the distribution of the adjacent pixels in the plain-image and its corresponding cipher-image. Figs. 3(a) and (b) show the correlation distribution of two horizontally adjacent pixels of the plain-image (Fig. 2(a)) and its ciphered image produced by the proposed scheme, respectively. Similar results can be obtained for vertically and diagonally adjacent pixels.

To further quantify and compare the correlations of adjacent pixels in the plain and cipher-image, the following procedure is carried out. First, randomly select 3000 pairs of adjacent pixels in each direction from the plain-image and its ciphered image. Then, calculate the correlation coefficient $r_{x,y}$ of each pair by using the following three formulas:



(a) (b)

Figure 3. Correlation of two horizontally adjacent pixels in plain and cipher-image. (a) plain-image. (b) cipher-iamge.

$$r_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2\right)\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2\right)}}, \tag{9}$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{10}$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i, \tag{11}$$

where x_i and y_i are grayscale values of the i^{th} pair of adjacent pixels, and N denotes the total number of samples.

The results of the correlation coefficients of adjacent pixels in three directions for the plain-image and its ciphered image are given in Table 1. It's clear from Fig. 3 and Table 1 that the strong correlation between adjacent pixels in plain-image is effectively erased in the cipher-image produced by the proposed scheme.

TABLE I. CORRELATION COEFFICIENTS OF TWO ADJACENT PIXELS IN TWO IMAGES

Direction	Plain-image	Cipher-image
Horizontal	0.9383	−0.0104
Vertical	0.9396	−0.0189
Diagonal	0.8996	−0.0219

B. Key space analysis

The key space is the total number of different keys that can be used in the encryption/decryption procedure. For an effective cryptosystem, the key space should be large enough to make brute-force attack infeasible. As mentioned above, the key of the proposed cryptosystem is composed of two parts: permutation key *Key-P* and diffusion key *Key-D*. As mentioned above, the size of *Key-P* is $(N^2)^m$. *Key-D* consists of three floating point numbers (x_0, y_0, z_0, w_0). According to the IEEE floating-point standard, the computational precision of the 64-bit double-precision number is about 10^{-15}, so the total number of possible values of *Key-D* is approximately 10^{60}.

The two parts *key-P* and *key-D* are independent from each other. Therefore, the total key space *Key-S* is

$$Key\text{-}S = key\text{-}P(N, m) \times key\text{-}D \approx (N^2)^m \times 2^{200}. \tag{12}$$

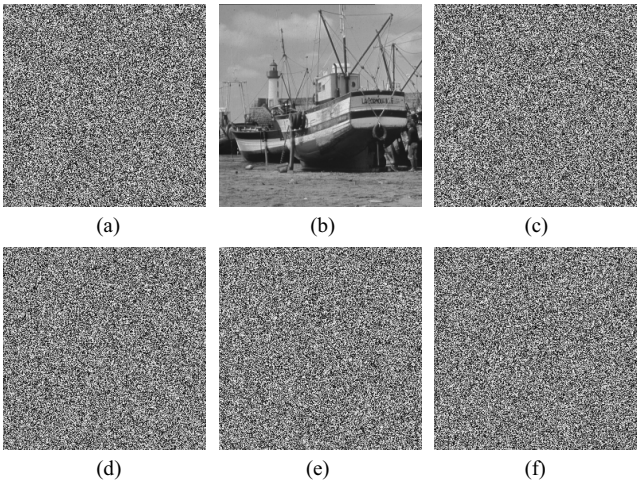
Here, it is proposed to take $m = 2$. If $N \geq 256$, the total size satisfies

$$Key\text{-}S \geq 2^{232}, \tag{13}$$

which is large enough to make exhaustive search infeasible.

C. Key Sensitivity Analysis

To evaluate the key sensitivity of the proposed scheme, the plain-image is firstly encrypted using the diffusion key (x_0=−3.7, y_0=−4.1, z_0=6.6, w_0=5.8) and the resultant cipher-image is shown in Fig. 4(a). Then the ciphered image is tried to be decrypted using five decryption keys: (i) (x_0=−3.7, y_0=−4.1, z_0=6.6, w_0=5.8), (ii) (x_0=−3.69999999999999, y_0=−4.1, z_0=6.6, w_0=5.8), (iii) (x_0=−3.7, y_0=−4.09999999999999, z_0=6.6, w_0=5.8), (iv) (x_0=−3.7, y_0=−4.1, z_0=6.60000000000001, w_0=5.8) and (v) (x_0=−3.7, y_0=−4.1, z_0=6.6, w_0=5.80000000000001). The resultant decrypted images are shown in Fig. 4(b), (c), (d), (e) and (f), respectively. As can be seen from Fig. 4, even an almost perfect guess of the key does not reveal any information about the plain-image. Therefore, the proposed scheme fully satisfies the key sensitivity requirement.



(a) (b) (c)

(d) (e) (f)

Figure 4. Results of key sensitivity test. (a) ciphered image using key (x_0=−3.7, y_0=−4.1, z_0=6.6, w_0=5.8). (b) deciphered image using key (x_0=−3.7, y_0=−4.1, z_0=6.6, w_0=5.8). (c) deciphered image using key (x_0=−3.69999999999999, y_0=−4.1, z_0=6.6, w_0=5.8). (d) deciphered image using key (x_0=−3.7, y_0=−4.09999999999999, z_0=6.6, w_0=5.8). (e) deciphered image using key (x_0=−3.7, y_0=−4.1, z_0=6.60000000000001, w_0=5.8). (f) deciphered image using key (x_0=−3.7, y_0=−4.1, z_0=6.6, w_0=5.80000000000001).

V. CONCLUSIONS

This paper has proposed a security improved symmetric image cipher to address the security problems encountered by many existing chaos-based image cryptosystems. The cat map and hyperchaotic Lorenz system are employed to decorrelate the relationship among adjacent pixels and confuse the relationship between cipher image and plain-image, respectively. Compared with ordinary chaotic systems, hyperchaotic systems have more complex dynamical behaviors and number of system variables. Therefore, the proposed cryptosystem has stronger

unpredictability and larger key space. Moreover, in diffusion stage, the key stream elements quantified from each iteration of hyperchaotic Lorenz system are reordered according to the previous plain-pixel. As a result, the quantified key stream is related not only to the key but also to the plain-image, which further improves the security against known/chosen-plaintext attack. Both theoretical analyses and experimental results indicate the proposed image cryptosystem has a high level of security. Therefore, the proposed scheme has excellent potential for practical image encryption applications.

ACKNOWLEDGMENT

This work was supported by the sub-program (NO. 2011BAK03B05-4, Research on emergency communications technology and equipment for fire fighting of major disasters) of National Science & Technology Pillar Program during the 12th Five-year Plan Period (NO. 2011BAK03B05, Research on new equipment and application technology for city fire fighting).

REFERENCES

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

[2] Chen GR, Mao YB, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solitons & Fractals, 21(3), 749-761, 2004.

[3] Lian SG, Sun JS, Wang ZQ. A block cipher based on a suitable use of the chaotic standard map. Chaos Solitons & Fractals, 26(1), 117-129, 2005.

[4] Pareek NK, Patidar V, Sud KK. Image encryption using chaotic logistic map. Image and Vision Computing, 24(9), 926-934, 2006.

[5] Kwok HS, Tang WKS. A fast image encryption system based on chaotic maps with finite precision representation. Chaos Solitons & Fractals, 32(4), 1518-1529, 2007.

[6] Behnia S, Akhshani A, Ahadpour S, et al. A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps. Physics Letters A, 366(4-5), 391-396, 2007.

[7] Wong KW, Kwok BSH, Law WS. A fast image encryption scheme based on chaotic standard map. Physics Letters A, 372(15), 2645-2652, 2008.

[8] Patidar V, Pareek NK, Sud KK. A new substitution-diffusion based image cipher using chaotic standard and logistic maps. Communications in Nonlinear Science and Numerical Simulation, 14(7), 3056-3075, 2009.

[9] Rhouma R, Meherzi S, Belghith S. OCML-based colour image encryption. Chaos Solitons & Fractals, 40(1), 309-318, 2009.

[10] Wang Y, Wong KW, Liao XF, et al. A chaos-based image encryption algorithm with variable control parameters. Chaos Solitons & Fractals, 41(4), 1773-1783, 2009.

[11] Fu C, Chen JJ, Zou H, et al. A chaos-based digital image encryption scheme with an improved diffusion strategy. Optics Express, 20(3), 2363-2378, 2012.

[12] Alvarez G, Li SJ. Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption. Communications in Nonlinear Science and Numerical Simulation, 14(11), 3743-3749, 2009.

[13] Li CQ, Li SJ, Asim M, et al. On the security defects of an image encryption scheme. Image and Vision Computing, 27(9), 1371-1381, 2009.

[14] Solak E, Rhouma R, Belghith S. Cryptanalysis of a multi-chaotic systems based image cryptosystem. Optics Communications, 283(2), 232-236, 2010.

[15] Gao TG, Chen GR, Chen ZQ, et al. The generation and circuit implementation of a new hyper-chaos based upon Lorenz system. Physics Letters A, 361(1-2), 78-86, 2007.

[16] Jia Q. Hyperchaos generated from the lorenz chaotic system and its control. Physics Letters A, 366(3), 217-222, 2007.