A Wheel-Switch Selective Image Encryption Scheme Using Spatiotemporal Chaotic System

Xing-Yuan Wang[a] and Xue-Mei Bao[a]

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, 116024, China

Reprint requests to X.-Y. W.; E-mail: wangxy@dlut.edu.cn

Z. Naturforsch. **69a,** 61 – 69 (2014) / DOI: 10.5560/ZNA.2013-0075
Received June 23, 2013 / revised September 21, 2013 / published online December 4, 2013

In this paper, we propose a novel selective image encryption scheme using a one-way coupled map lattice (CML) consisting of logistic maps and a selector constructed by two variants of a cyclic shift register (VCSR). The initial conditions and the coupling constant of CML in our scheme are influenced by all the contents of the plain image. Moreover, the selector is closely related to the nonencrypted part of the plain image. In addition, we select only a portion of image data to encrypt via a wheel-switch scheme governed by the selector. Users can select an appropriate proportion to encrypt the plain image for their different demands of security and efficiency. Experimental results and theoretical analysis show that the cryptosystem is effective and can resist various typical attacks.

Key words: Image Encryption; Selective Encryption; Spatiotemporal Chaotic System.

1. Introduction

With the rapid development of computer network technology, a lot of sensitive information is transmitted over the network, and information security becomes more and more important. Image encryption is different from text encryption due to some inherent features such as bulk data capacity and high correlation among pixels. Therefore, traditional cryptographic techniques such as data encryption algorithm (DES), international data encryption algorithm (IDES), and RSA (a public-key cryptography, proposed by Rivest, Shamir, and Adleman) are no longer suitable for image encryption. Chaotic systems have several significant features, such as sensitive dependence on initial conditions, pseudo-randomness and ergodicity [1]. These features characterize good properties of diffusion and confusion, which make it very suitable for image encryption. Lots of image encryption methods are proposed using chaotic systems in this era [2 – 7].

However, up to now, most of them are proved to be insecure [8 – 13]. The most serious problem in individual chaotic systems is that the chaotic dynamics degrade rapidly when they are realized with finite precisions in digital computers [14]. Coupled map lattice (CML) based spatiotemporal chaotic systems possess excellent chaotic dynamical properties and could maintain much longer periodicity [15], which as a result are widely used in chaotic cryptography in recent years [5, 16 – 20]. In our scheme, the keystream is generated from the CML.

In order to achieve a good tradeoff between security and efficiency, selective encryption, in which only partial image data are encrypted, has been introduced to image encryption [21 – 25]. Compared to traditional encryption schemes, the speed of selective encryption scheme is faster than that of them using the same or similar encryption algorithm. In our scheme, we select only a portion of image data to encrypt. Experiment results and security analysis show that the scheme not only can achieve good encryption results, but also can resist against common attacks.

The remaining of the paper is organised as follows. In Section 2, the CML and the selector used in the proposed algorithm are introduced, and in Section 3, the encryption and decryption algorithms are described. Section 4 provides simulation results. Performance analysis is given in Section 5. Finally, this paper is concluded in Section 6.

2. Preliminary Materials

2.1. Description of CML

CML is a dynamical system with discrete-time and discrete-space. It consists of nonlinear maps called as local maps on the lattice sites. Each local map is coupled with other local maps governed by certain coupling rules. Because of the intrinsic nonlinear dynamics of each local map and the diffusion due to the spatial coupling among the local maps, a CML exhibits spatiotemporal chaos [16]. CML is described as

$$x_{n+1}(j) = (1-\varepsilon)\tau(x_n(j)) + \varepsilon\tau(x_n(j-1)), \quad j = 1,2,\ldots,L, \tag{1}$$

where n is the time index, j is the lattice site index, $\varepsilon \in (0,1)$ is the coupling constant, and L is the lattice length. $x_n(j)$ represents the state variable for the jth site at time n. The periodic boundary conditions $x_n(j) = x_n(L-j)$ for any valid j are used in the CML. $\tau(x)$ is a local map given by

$$\tau(x) = \mu x(1-x), \quad x \in [0,1], \quad \mu \in [0,4], \tag{2}$$

which is chaotic when $\mu > 3.57$. μ and L are both set to 4 throughout this paper.

2.2. Description of the Selector

The selector used in our scheme is constructed with two variants of cyclic shift registers (VCSRs). The cyclic shift register (CSR) is a sequence of bits. Each time a bit is needed, all of the bits in the CSR are shifted one bit to the right, and simultaneously the right-most bit is shifted to the left-most position. Figure 1 shows the VCSR used in our scheme. As can be seen from this figure, before shifting, the right-most bit b_1 is replaced by the XORed result of a particular bit and itself, otherwise, the VCSR is identical to a CSR. The selector is used to govern the wheel-switch scheme according to the two bits produced from two VCSRs. In the following, we will introduce the rule to obtain the particular bit and how the selector governs the wheel-switch scheme in detail.

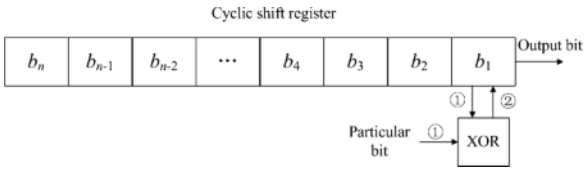

Fig. 1. Variants of a cyclic shift register (VCSR).

3. The Proposed Image Cryptosystem

3.1. Selective Image Encryption

We know a bit can contain different amounts of information depending on its position in the pixel [26]. For example, a "1" at the 8th bit of a pixel represents 128 (2^7), but it only represents 1 (2^0) at the first bit. The percentage of information $p(i)$ provided by the ith bit is given by

$$p(i) = \frac{2^{i-1}}{\sum_{j=1}^{8} 2^{j-1}}, \quad i = \{1,2,\ldots,8\}. \tag{3}$$

$p(i)$ is shown in Table 1. We can find that the higher 4 bits (8th, 7th, 6th, and 5th) carry 94.125% of the total information of the image, but the lower 4 bits (4th, 3rd, 2nd, and 1st) carry less than 6% of the image information.

Moreover, we select the image "Lena" of size 256×256 with 256 grey levels shown in Figure 2a, divide it into 8 bitplain images according to the bit locations within a pixel. Figure 2b – i show these 8 bitplain images obtained by collecting the 8th bit to the 1st bit of all the pixels, respectively. It is clearly revealed that the higher bits portray the skeleton of "Lena", as illustrated in Figure 2b – e, while those lower ones just look like random noise shown in Figure 2f – i. The visibility of the image gradually degrades with the decreasing of bit index.

Based on this fact, we choose to encrypt only a portion of higher bits (8th, 7th, 6th, and 5th or more) in

Table 1. Percentage of pixel information contributed by different bits.

Bit position i in the pixel	Percentage $p(i)$ of the pixel information (%)
1	0.3322
2	0.7843
3	1.5686
4	3.1373
5	6.275
6	12.55
7	25.10
8	50.20

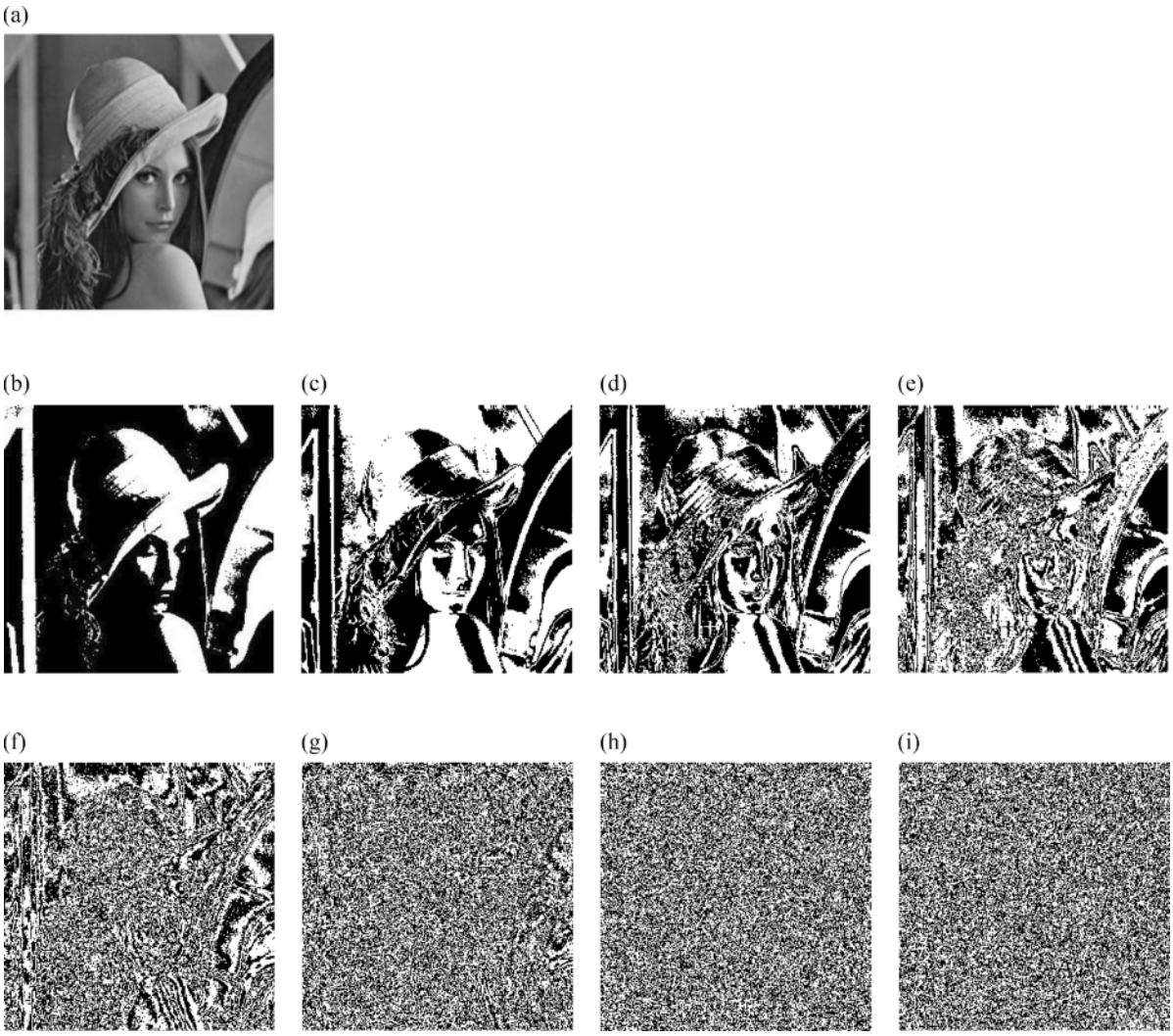

Fig. 2. Bitplane images of "Lena". (b) – (i) The images whose pixel value is only composed of 8th bit to the 1st bit, respectively.

each pixel. As these bits depict the skeleton of an image, if they are encrypted, the whole image will become unrecognizable. How many bits are selected to encrypt is decided by users for their different demands of security and efficiency.

3.2. Encryption Scheme

Confusion and diffusion processes in cryptography proposed by Shannon [27] have been applied in image encryption successfully. Most of the encryption schemes are composed of both of the two processes. However, in our cryptosystem we omit the confusion phase, just as the flowchart of our encryption process shown in Figure 3. The diffusion process is implemented via a wheel-switch scheme governed by a selector.

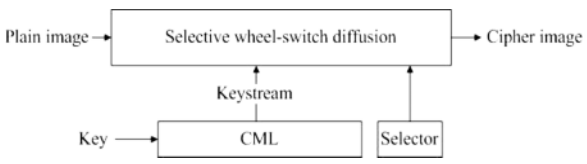

Fig. 3. Flowchart of the encryption process.

In more details, for an $M \times N$ plain image P, the encryption process maybe summarized as follows:

Step 1: Let $y_0(1)$, $y_0(2)$, $y_0(3)$, $y_0(4)$, and σ serve as the secret keys. We calculate the average value δ (serving as the secret key) of all the pixels in P, which is used to generate the initial conditions $x_0(1)$, $x_0(2)$, $x_0(3)$, $x_0(4)$, and the coupling constant ε of CML by using the following formulas:

$$\eta = \delta/256, \tag{4}$$
$$x_0(j) = (y_0(j) + \eta) \times 0.5, \quad (1 \le j \le 4), \tag{5}$$
$$\varepsilon = (\sigma + \eta) \times 0.5. \tag{6}$$

Initialize CML with $x_0(j)$ $(1 \le j \le 4)$ and ε, then iterate CML 500 $+M \times N$ times, discard the former 500 values to avoid harmful effect. Each lattice generates $M \times N$ values, so the chaotic system generates $M \times N \times 4$ values totally.

Step 2: Let $P(i)$ denote the ith $(1 \le i \le M \times N)$ pixel value of P. To encrypt $P(i)$, we write it in a binary representation:

$$P(i) = p_8 p_7 \ldots p_k \ldots p_1, \quad p_k \in \{0,1\}. \tag{7}$$

We select v $(4 \le v \le 7$ and serves as the secret keys) higher bits, e.g., through p_8 to p_{8-v+1} from each pixel.

In order to facilitate the following discussion, we use $P^m(i)$ to denote the m higher bits of $P(i)$ as given by

$$P^m(i) = p_8 p_7 \ldots p_{8-m+1}, \tag{8}$$

and let $P_n(i)$ denote the nth bit of $P(i)$, that is p_n.

Step 3: Let $x_i(j)$ $(1 \le i \le M \times N, 1 \le j \le 4, 0 \le x_i(j) \le 1)$ denote the output value of CML for the jth lattice at time i. Transform $x_i(j)$ into integers $x'_i(j) \in [0, 2^v - 1]$ and $x''_i(j) \in [0, v-1]$ as follows:

$$x'_i(j) = \lfloor x_i(j) \times 10^{14} \rfloor \bmod 2^v, \tag{9}$$
$$x''_i(j) = x'_i(j) \bmod v, \tag{10}$$

where "$x \bmod y$" returns the remainder of x divided by y, $\lfloor x \rfloor$ rounds x to the nearest integer less than or equal to x.

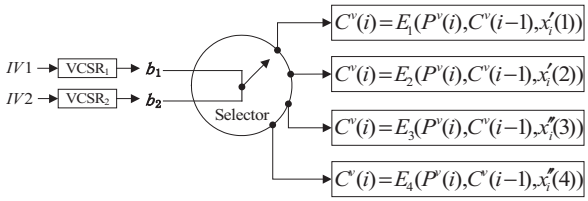

Fig. 4. Sketch diagram of wheel-switch scheme.

Step 4: Use $x'_i(j)$ and $x''_i(j)$ to diffuse $P^v(i)$ via a wheel-switch scheme governed by the selector, the sketch diagram of which is illustrated in Figure 4.

As shown in Figure 4, initial vectors, i.e., IV_1 and IV_2, must be set properly and serve as the secret keys. IV_1 is the initial vector of VCSR_1, IV_2 is the initial vector of VCSR_2. The highest bit of the nonencrypted part of $P(i)$, namely, $P_{8-v}(i)$ serves as the particular bit of VCSR_1 and VCSR_2, just as we can see in Figure 1. The wheel-switch scheme is not complicated. First, VCSR_1 and VCSR_2 produce one bit of the sequence, respectively, say b_1 and b_2. According to the values of b_1 and b_2, select different functions, e.g., E_1, E_2, E_3, and E_4 to produce the ith ciphered value $C^v(i)$, which is corresponding to $P^v(i)$, that is:

(i) If $b_1 = 0$ and $b_2 = 0$, then

$$\begin{aligned} C^v(i) &= E_1(P^v(i), C^v(i-1), x'_i(1)) \\ &= (P^v(i) + C^v(i-1) + x'_i(1)) \bmod 2^v. \end{aligned} \tag{11}$$

(ii) If $b_1 = 0$ and $b_2 = 1$, then

$$\begin{aligned} C^v(i) &= E_2(P^v(i), C^v(i-1), x'_i(2)) \\ &= P^v(i) \oplus C^v(i-1) \oplus x'_i(2), \end{aligned} \tag{12}$$

where $\oplus$ represents XOR operation.

(iii) If $b_1 = 1$ and $b_2 = 0$, then

$$\begin{aligned} C^v(i) &= E_3(P^v(i), C^v(i-1), x''_i(3)) \\ &= \mathrm{ROR}(P^v(i) \oplus C^v(i-1), x''_i(3)), \end{aligned} \tag{13}$$

where $\mathrm{ROR}(a,b)$ performs the b-bit right cyclic shift on the binary sequence a.

(iv) If $b_1 = 1$ and $b_2 = 1$, then

$$\begin{aligned} C^v(i) &= E_4(P^v(i), C^v(i-1), x''_i(4)) \\ &= \mathrm{ROL}(P^v(i) \oplus C^v(i-1), x''_i(4)), \end{aligned} \tag{14}$$

where $\mathrm{ROL}(a,b)$ performs the b-bit left cyclic shift on the binary sequence a.

What should be noted is that $C^v(0) = 0$. Repeat Step 4 until each $P^v(i)$ is encrypted.

After diffusions, $C^v(i)$ serves as the v higher bit of $C(i)$ and the nonencrypted part of $P(i)$ as the lower part; $C(i)$ is the final ciphered pixel.

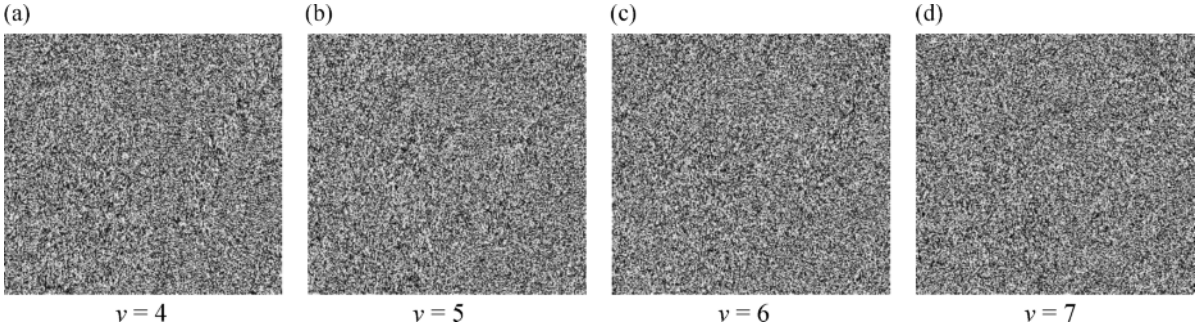

Fig. 5. Ciphered images of "Lena" when v takes different values.

3.3. Decryption Scheme

The decryption procedure is reverse to that of the encryption procedure illustrated above.

We can obtian $P^v(i)$ from $C^v(i)$ and $C_{8-v}(i)$ following Step 4 of the encryption phase, except that we should use (15) – (18) to replace (11) – (14) and that the particular bit of VCSR_1 and VCSR_2 is $C_{8-v}(i)$:

$$\begin{aligned} P^v(i) &= E_1^{-1}(C^v(i), C^v(i-1), x_i'(1)) \\ &= (C^v(i) - C^v(i-1) - x_i'(1)) \bmod 2^v , \end{aligned} \tag{15}$$

$$\begin{aligned} P^v(i) &= E_2^{-1}(C^v(i), C^v(i-1), x_i'(2)) \\ &= C^v(i) \oplus C^v(i-1) \oplus x_i'(2) , \end{aligned} \tag{16}$$

$$\begin{aligned} P^v(i) &= E_3^{-1}(C^v(i), C^v(i-1), x_i''(3)) \\ &= \mathrm{ROL}(C^v(i), x_i''(3)) \oplus C^v(i-1) , \end{aligned} \tag{17}$$

$$\begin{aligned} P^v(i) &= E_4^{-1}(C^v(i), C^v(i-1), x_i''(4)) \\ &= \mathrm{ROR}(C^v(i), x_i''(4)) \oplus C^v(i-1) . \end{aligned} \tag{18}$$

4. Simulation Results

We have used Microsoft Visual C++ 6.0 to run the encryption and decryption programs in a personal computer with a Pentium 4 CPU 1.70 GHz, 256 MB RAM and 60 GB hard-disk capacity, and the operation system is Microsoft Windows XP. Our simulation results are shown in Figure 5. The 256×256 grey-scale image "Lena" sized 65.0 kB (Fig. 2a) is used as the plain image. Figure 5a – d show the ciphered images when v takes different values. From Figure 5, it is clear that the ciphered images are all unrecognized with v being equal to 4, 5, 6, and 7.

5. Performance Analysis

A good encryption scheme should be robust against all kinds of cryptanalytic, statistical, and brute-force attacks and have higher efficiency. Some performance analysis has been performed on the proposed image encryption scheme.

5.1. Key Space Analysis

The size of the key space characterizes the capability of resisting brute-force attack. A short key means that the best encryption algorithm can be broken by exhaustive search (also known as brute-force attacks) in a reasonable amount of time, while the reverse is not true. In our algorithm, IV_1 (32 bits), IV_2 (32 bits), v, $y_0(1)$, $y_0(2)$, $y_0(3)$, $y_0(4)$, δ, and σ are used as secret keys. The key space is large enough for common applications to resist brute-force attack.

5.2. Histogram Analysis

The distribution of the ciphered image is a major concern. A histogram is often defined as a graph that shows the distribution of pixel values of an image. If it is not flat enough, certain amount of information can be guessed by the statistical attack opponent. This makes cipher-only attack easier through analyzing the statistic property of ciphered image. Therefore, a flat distribution is desirable in cryptography. Figure 6 illustrates the histograms of ciphered images when v takes different values.

It is clear from Figure 6 that the proposed scheme results in very flat distributions of ciphered images, which can resist cipher-only attack.

5.3. Information Entropy Analysis

Information entropy is the most important feature of randomness. Let m be the information source then the

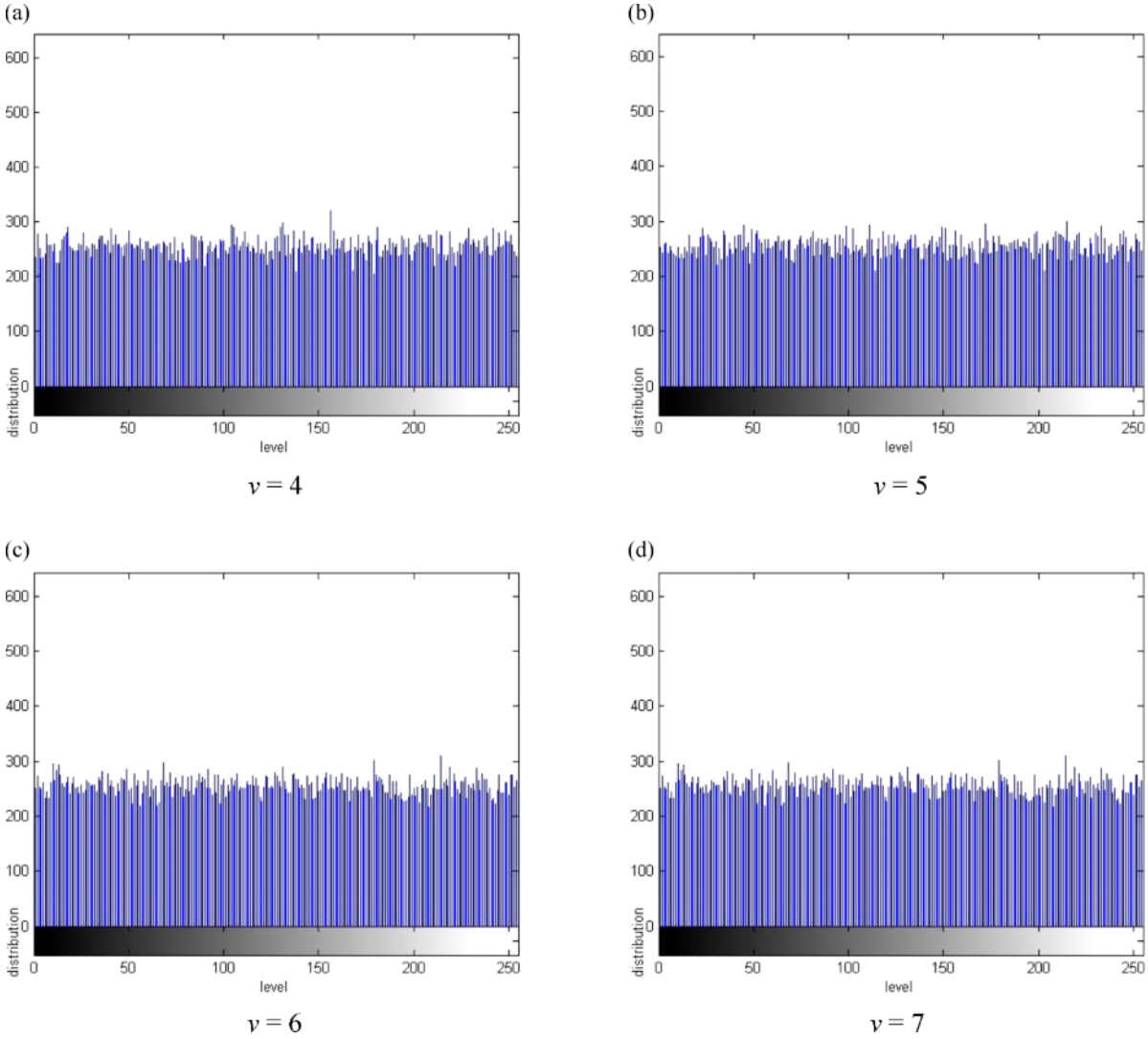

Fig. 6 (colour online). Histograms of ciphered images when v takes different values.

formula for calculating information entropy is

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) \log_2 \frac{1}{p(m_i)}, \tag{19}$$

where $p(m_i)$ represents the probability of symbol m_i. Assume that there are 2^8 states of the information source and they appear with the same probability. According to (19), we can get the ideal $H(m) = 8$, which shows that the information is completely random. The information entropy of the ciphered image should be close to 8 after encryption. The more it gets close to 8, the less possible for the scheme to divulge information. Information entropy of the plain image is 7.5683. The test results of information entropy of the ciphered images when v takes different values are presented in Table 2.

Table 2. Information entropy of the ciphered images when v takes different values.

v	$H(m)$
4	7.9971
5	7.9973
6	7.9974
7	7.9979

The results are very close to 8 with v being equal to 4, 5, 6, and 7; we can conclude that the ciphered images using the proposed scheme could hardly divulge information for any eligible v.

5.4. Correlation Analysis

Correlation indicates the strength and direction of a linear relationship between two random variables. In image processing, it is usually employed to investigate the relationship between two adjacent pixels. The correlation between adjacent pixels is usually high in a recognizable image. The less the correlation of two adjacent pixels is, the safer the image is. In order to investigate the confusion effect of ciphered images, the correlation coefficients between two horizontally adjacent pixels, two vertically pixels and two diagonally adjacent pixels are tested, respectively. The following equation calculates the correlation coefficient of two adjacent pixels:

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{20}$$

where

$$\begin{aligned} \operatorname{cov}(x,y) &= \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \\ E(x) &= \frac{1}{N}\sum_{i=1}^{N}x_i, \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2. \end{aligned} \tag{21}$$

As there are three kinds of adjacent directions as mentioned above, we randomly select 2000 pairs of adjacent pixels in each direction from the plain image and its ciphered images when v takes different values. Their correlation coefficients are then calculated. The correlation coefficients of the plain image are horizontally 0.9743, vertically 0.9300, and diagonally 0.9502, respectively. The correlation coefficients of the ciphered images are listed in Table 3. From the results, we know that there is a strong correlation between adjacent pixels of each direction in the plain image since the correlation coefficients are all greater than 0.9. While in the ciphered images, these values are all smaller than 0.1 for any eligible v, which indicate that the ciphered images encrypted using the proposed scheme are safe with a negligible correlation between adjacent pixels.

Table 3. Correlation coefficients of the ciphered images when v takes different values.

v	Horizontally	Vertically	Diagonally
4	0.0481	−0.0030	−0.0063
5	0.0301	0.0108	−0.0104
6	0.0109	0.0090	0.0057
7	0.0003	0.0016	0.0034

5.5. NPCR and UACI Analysis

The abbreviation NPCR stands for the number of pixels change rate while one pixel of the plain image changes. The more NPCR gets close to 100%, the more sensitive is the cryptosystem to the changing of plain image, and the more sensitive for the cryptosystem to resist plaintext attack. The unified average changing intensity (UACI) stands for the average intensity of differences between the plain image and the ciphered image. The bigger UACI is, the more sensitive for the cryptosystem to resist differential attack. Here are the formulas to calculate NPCR and UACI:

$$\mathrm{NPCR} = \frac{\sum_{ij} D(i,j)}{W \times H} \times 100\%, \tag{22}$$

$$\mathrm{UACI} = \frac{1}{W \times H}\left[\sum_{i,j}\frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\%, \tag{23}$$

where W and H represent the width and height of the image, respectively, C_1 and C_2 are respectively the ciphered images before and after one pixel of the plain image is changed. For the pixel at position (i,j), if $C_1(i,j) \neq C_2(i,j)$, let $D(i,j) = 1$; else let $D(i,j) = 0$. In our test, only a lowest bit of one pixel of the plain image is changed. The test results of NPCR and UACI when v takes different values are shown in Table 4.

As depicted in Table 4, NPCRs are all greater than 90% and UACI are all greater than 33%, indicating that

Table 4. Number of pixels change rate (NPCR) and unified average changing intensity (UACI) when v takes different values.

v	NPCR (%)	UACI (%)
4	93.74	33.42
5	96.84	33.46
6	98.44	33.44
7	99.36	33.50

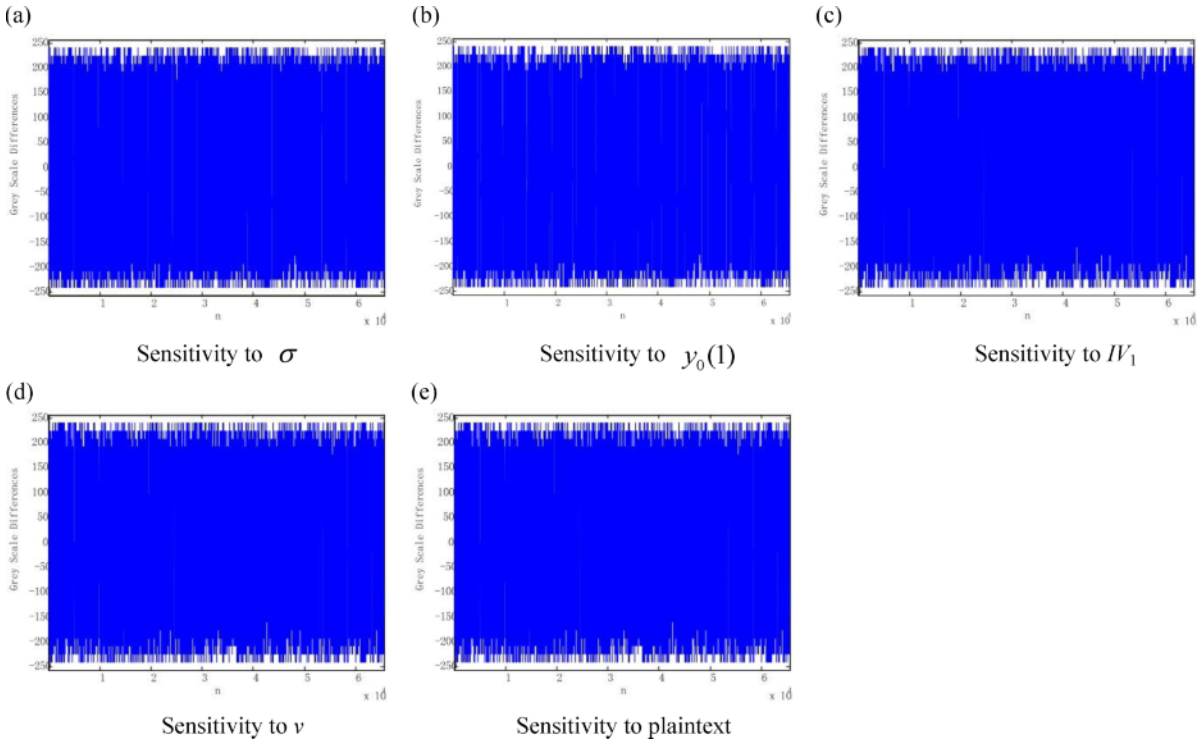

Fig. 7 (colour online). Sensitivity test results.

the proposed cryptosystem could resist plaintext attack and differential attack effectively. Moreover, from Table 4 we can see that the larger v is, the larger NPCR is and the safer the cryptosystem is. Therefore, users can choose appropriate v for their different demand of security.

5.6. Sensitivity

An excellent cryptosystem should be sensitive to the secret keys as well as the plaintext. In order to test the sensitivity of our encryption scheme, several tests have been performed. We set the secret keys as: $\mathrm{IV}_1 = 01001010111010101110101011001011$, $\mathrm{IV}_2 = 10100101010010010011010011001101$, $v = 4$, $\sigma = 0.87965$, $\delta = 98.1778$, $y_0(1) = 0.31456$, $y_0(2) = 0.11582$, $y_0(3) = 0.91802$, and $y_0(4) = 0.65321$. The test results of the differences between two ciphered images are shown in Figure 7. Figure 7a shows the result when σ is changed to 0.879650000000001; Figure 7b shows the result when $y_0(1)$ is changed to 0.314560000000001; Figure 7c shows the result when the left-most bit of IV_1 is changed from 0 to 1; Figure 7d shows the result when v is changed from 4 to 5. From Figure 7a–d, we can see that the proposed encryption scheme is sensitive to the secret keys. Figure 7e shows the differences between two ciphered images when a lowest bit of the pixel value of the plain image is changed.

Our scheme is sensitive to the plaintext, because on one hand the initial conditions $x_0(1)$, $x_0(2)$, $x_0(3)$, and $x_0(4)$ and the coupling constant ε of CML are influenced by all the contents of the plain image by using (4)–(6) , on the other hand the selector is closely related to the nonencrypted part of the plain image. The high sensitivity to plaintext ensures the cryptosystem could resist chosen plaintext attack.

5.7. Cost and Speed Analysis

The proposed algorithm is easy to realize. The time used for encryption on the image “Lena” when v takes different values is listed in Table 5.

Table 5 indicates that the encryption speeds when v takes different values are all very fast. In addition, the smaller v is, the faster the encryption speed is. Users

Table 5. Time used for encryption when v takes different values.

v	Time (s)
4	0.016
5	0.031
6	0.036
7	0.047

can select appropriate v for their different demands of security and efficiency.

6. Conclusion

This paper proposes a novel selective image encryption system using CML and a selector. We can see that the proposed cryptosystem can process any size of image. Users can select appropriate proportion to encrypt for their different demands of security and efficiency. Experiment results and security analysis show that the scheme not only can achieve good encryption results and large key space, but also can resist against common attacks.

Acknowledgement

This research is supported by the National Natural Science Foundation of China (Nos: 61370145, 61173183, and 60973152), the Doctoral Program Foundation of Institution of Higher Education of China (No: 20070141014), the National Natural Science Foundation of Liaoning province (No: 20082165), and the Fundamental Research Funds for the Central Universities (No: DUT12JB06).

[1] B. Liu, J. Peng, Nonlinear Dynamics, High Education Press, Beijing 2004.
[2] M. Françoisa, T. Grosgesa, D. Barchiesia, and R. Errab, Image Commun. **27**, 249 (2012).
[3] X. Wang and Q.Yu, Commun. Nonlin. Sci. Numer. Simul. **14**, 574 (2009).
[4] Z. Liu, H. Zhang, and Q. Zhang, IEEE Trans. Neural Networks **21**, 1710 (2010).
[5] X. Wang and L. Teng, Nonlin. Dyn. **67**, 365 (2012).
[6] Y. Liu and Y. Zheng, Nonlin. Dyn. **57**, 431 (2009).
[7] X. Wang, L. Yang, and R. Liu, Nonlin. Dyn. **62**, 615 (2010).
[8] C. Cokal and E. Solak, Phys. Lett. A **373**, 1357 (2009).
[9] Y. Liu, C. Chen, G. Wen, and S. Tong, IEEE Trans. Neural Networks **22**, 1162 (2011).
[10] L. Zhao, A. Adhikari, and D. Xiao, Commun. Nonlin. Sci. Numer. Simul. **17**, 3303 (2011).
[11] J. Fu, H. Zhang, T. Ma, and Q. Zhang, Neurocomputing **73**, 795 (2010).
[12] T. Ma, W. B. Jiang, J. Fu, Y. Chai, L. P. Chen, and F. Z. Xue, Acta Physica Sinica **61**, 160506 (2012).
[13] C. Li and K. Lo, Signal Processing **91**, 949 (2010).
[14] D. D. Wheeler, Cryptologia **7**, 243 (1991).
[15] W. Liu, H. Lu, J. Kuang, and G. Hu, Int. J. Mod. Phys. B **18**, 2617 (2004).
[16] P. Li, Z. Li, W. A. Halang, and G. Chen, Chaos Solitons Fract. **32**, 1867 (2007).
[17] Y. Liu, S. Tong, W. Wang, and Y. Li, Int. J. Control Automation Systems **7**, 681 (2009).
[18] J. Fu, M. Yu, and T. Ma, Chinese Phys. B **20**, 120508 (2011).
[19] S. Lian, Chaos, Solitons Fract. **40**, 2509 (2009).
[20] X. Ge, F. Liu, B. Lu, and W. Wang, Phys. Lett. A **5**, 908 (2011).
[21] T. Xiang, K. W. Wong, and X. Liao, Chaos **17**, 023115 (2007).
[22] X. Wang and C. Jin, Opt. Commun. **285**, 412 (2012).
[23] T. Ma and J. Fu, Neurocomputing **74**, 857 (2011).
[24] H. Zhang, Y. Luo, and D. Liu, IEEE Trans. Neural Networks **20**, 1490 (2009).
[25] J. Liu, Pattern Reccognition **39**, 1509 (2006).
[26] H. Liu and X. Wang, Opt. Commun. **284**, 3895 (2011).
[27] C. E. Shannon, Bell System Techn. J. **28**, 656 (1949).