International Journal of Security and Its Applications
Vol.8, No.6 (2014), pp.61-70
http://dx.doi.org/10.14257/ijsia.2014.8.6.06

A Novel Image Encryption Scheme based on the LSM Chaotic System

Congxu Zhu[1], Yuping Hu[2] and Xinran Zhou[1]

[1]*School of Information Science and Engineering, Central South University, Changsha, 410083, China*
[2]*School of Informatics, Guangdong University of Finance & Economics, Guangzhou 510320, China*
zhucx@csu.edu.cn, okhyp@gdufe.edu.cn, zhou_xinran@126.com

Abstract

The Logistic-Sine map (LSM) chaotic system is introduced by combing the Logistic map (LM) and Sin map (SM). The bifurcation diagrams show that the chaotic range of LSM is much larger than these of the Logistic or Tent maps. Complexity characteristics of Logistic map, Sine map and Logistic-Sine map are analyzed based on C_0 algorithm. The results show that C_0 complexity value of the LSM is the largest one among the three. Then, a novel image encryption scheme based on the LSM chaotic system was proposed. First, the positions of image pixels are shuffled through swapping positions randomly by using chaotic values. The permutation sequences are ralated to plain-images by introducing the plaintext feedback technique. Second, the diffusion procedure with LSM is introduced to diffuse the image, which is composed of two rounds. The experimental results and analysis by using several security measures show that the proposed image encryption scheme has high security and efficiency.

Keywords: *image encryption, Logistic-sine map, permutation, diffusion*

1. Introduction

In the information era, digital images have been widely used for various applications, such as entertainment, business, health service and military affairs. Therefore, it is especially important to protect images from piracy. As a result, image encryption technology becomes an important issue of cryptography. However, bulk data size and high redundancy among the raw pixels of a digital image make the traditional encryption algorithms, such as DES, IDEA, AES, not able to be operated efficiently. Chaotic systems have many excellent intrinsic properties, such as ergodicity, sensitive to the initial condition and control parameters. These properties are analogous to the confusion and diffusion properties specified by Shannon [1]. Thus makes it natural to employ chaotic systems in image encryption algorithms [2–10].

Because one dimensional chaotic system have simple structures and are easy to implement [11–15]. Therefore, image encryption using one dimensional chaotic system will bring the cryptosystems high efficiency. But, 1D chaotic system also has some shortcomings in security because of their simplicity and small key space. Hence, it is important to enhance the chaotic performance of 1D chaotic system.

In many image encryption algorithms, the confusion and diffusion processes proposed by Shannon [1] are adopted. These processes include a permutation-diffusion structure. While many proposed chaotic image encryption system adopted Arnold cat

ISSN: 1738-9976 IJSIA

map to shuffle the positions of the pixels by confuse phase [7-9]. In many proposed chaotic image encryption system, the methods of permuting positions of image pixels are based on sorting order the chaotic sequence [16]. To our best knowledge, current image encryption algorithms have following flaws more or less: Arnold transform's short-cycle problem. The permutation process was separated from diffusion process. The permutation sequences have nothing to do with plain-images. There are strict limits on the image, such as the requirement that the image must be square. These flaws of current image encryption algorithms limited their scope of application or bring down their security.

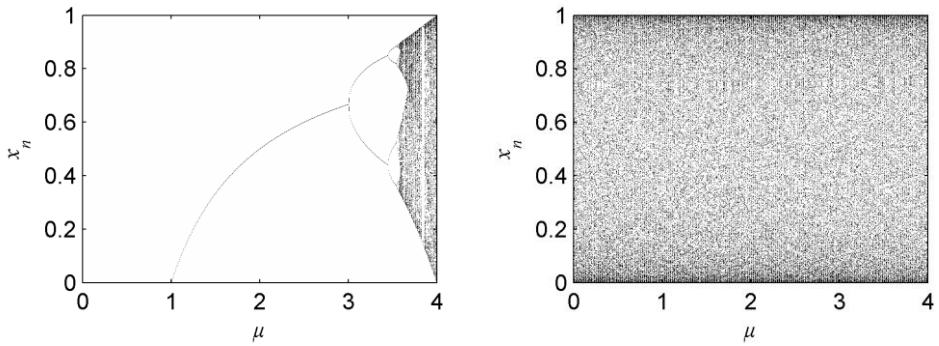
To address these above-mentioned problems, this paper introduces a new 1D chaotic system with a higher complexity and a simple structure. By using the new 1D chaotic system, a novel permutation–diffusion image encryption algorithm is proposed. In the permutation part, we develop different permutation sequences for different plain-images by means of mapping some information of the plain-image to the generation process of the permutation sequence. Thus makes the permutation behave in a ''one time pad'' manner. In the diffusion part, another feedback technique is employed to make the equivalent key generation depend on both the plain-image and the chaotic values. By combining the proposed permutation and diffusion technique, the scheme frustrates the known attacks [17-18]. In addition, we adopt two runs of diffusion process to make the scheme sensitive to changes of plain-image and initial keys.

2. The LSM Chaotic System

Based on the Logistic and Sine chaotic maps, we introduce a new Logistic-Sine map (LSM), which can be denoted by Eq. (1):

$$x_{n+1} = \begin{cases} \mu x_n(1 - x_n) + \dfrac{(4-\mu)}{4}\sin(\pi x_n), & if\ x_{n+1} < 1 \\ \mu x_n(1 - x_n) + \dfrac{(4-\mu)}{4}\sin(\pi x_n) - 1, & if\ x_{n+1} > 1 \end{cases} \quad (1)$$

Where $x_n \in (0,1)$, $\mu \in (0, 4]$, μ is the system parameter. Figure 1 shows the bifurcation diagrams of Logistic map and LSM, Sine map has the diagram similar to Logistic. From Figure 1 one can see that LSM system is chaotic in the entire parameter range $\mu \in (0, 4]$. Namely, the LSM system has a wider chaotic range than the Logistic and Sine maps.



(a) Logistic chaotic system (b) LSM chaotic system

Figure 1. The Bifurcation Diagrams of the Chaotic System

The complexity analysis of chaotic sequences is a new topic at present [19]. A natural question to follow is whether the LSM chaotic system is more complex than the Logistic and Sine maps. As a new method, C_0 complexity is used to compute the complexities of chaotic systems [20]. Larger C_0 complexity values imply more complex. In this paper, we are using the C_0 complexity algorithm to measure the complexity of the chaotic Logistic, Sine and Logistic-Sine systems. Figure 2 shows the C_0 complexity with different values of system parameter μ. From Figure 2 one can see that the LSM chaotic system is more complex than the Logistic and Sine maps in a wider parameter range. This indicates that the LSM chaotic system has more excellent chaotic performance, and more suitable for applications in the field of secure communication and information encryption.

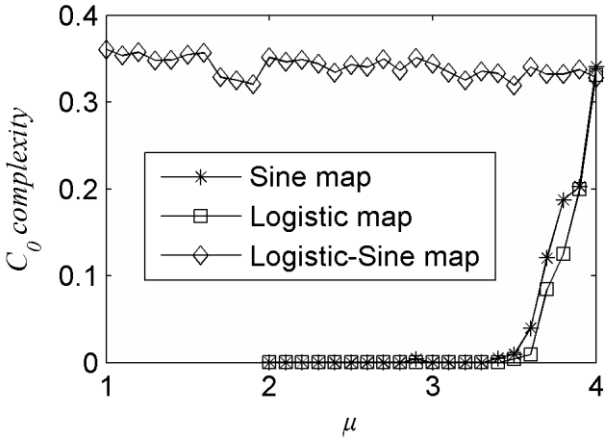


Figure 2. C_0 Complexity Versus System Parameter μ

3. The Proposed Cryptosystem

The plain-image to be encrypted is a 256 gray-scale image of size $L=M\times N$, it is an integer matrix of M rows N columns, in which the values range from 0 to 255. Its data can be treated as a one-dimensional vector $\mathbf{P}=\{p(1), p(2), \cdots, p(L)\}$. The proposed scheme includes the permutation process and two rounds of diffusion process. The permuted image pixels is denoted by $\mathbf{P}'=\{p'(1), p'(2), \cdots, p'(L)\}$. The cipher-image pixel sequence after the first round diffusion is denoted by $\mathbf{C}'=\{c'(1), c'(2), \cdots, c'(L)\}$, and the final cipher-image pixel sequence is denoted by $\mathbf{C}=\{c(1), c(2), \cdots, c(L)\}$. The initial secret keys include two groups of parameters (x_{10}, μ_1), (x_{20}, μ_2) and two integers C_0, K_0. Where $C_0 \in [1, 255]$, and $K_0 \in [1, 255]$.

3.1. Permutation Process

Using initial state value x_{10} and μ_1 of LSM, we firstly generate permutation sequence to change the position of the plain image pixel. Different from most proposed methods, our scheme is based on swapping positions randomly instead of sorting chaotic sequence. Suppose $\mathbf{T}=\{t(1),t(2),\ldots, t(L)\}$ is a permutation sequence generated in the permutation process, where $t(i)$ are integers, $t(i)\in [1, L]$ and $t(i) \neq t(j)$ if $i \neq j$. Our permutation scheme takes the following steps:

Step 1 Calculate the average value m_1 and the maximum m_2 of the plain-image $\{p(i)\}$. Note that $m_1=m_2$ if and only if $p(i)$ is a constant, where $i\in\{1,2,\ldots,L\}$. The permutation process can be simply ignored when this situation occurs during encryption. Otherwise, continue to execute Step 2 to Step 8.

Step 2 Let $x\leftarrow x_0\times m_1/m_2, \mu\leftarrow\mu_1$. Iterate the LSM by using Eq. (1) for N_0 times to get rid of transient effect, where N_0 is a constant. Initialize the permutation sequence $t(i)=i$, $i=1,2,\ldots,L$.

Step 3 Let $i\leftarrow 1$.

Step 4 To iterate the LSM for one times to obtain a new x, then computing a integer j by using current x according to the following formula:

$$j=\mathrm{mod}(\mathrm{floor}(x\times 10^{14}),L)+1. \quad (2)$$

Where floor(x) returns the nearest integer smaller than or equal to x, and mod(x, y) returns the remainder after division.

Step 5 Checking the values j, if $j=i$, then repeat step 4; else then go to step 6.

Step 6 Swapping the pixels in position i and j: $p'(i)\leftarrow p(j)$, $p'(j)\leftarrow p(i)$.

Step 7 Swapping the values $t(i)$ and $t(j)$.

Step 8 Let $i\leftarrow i+1$, return to Step 4 until i reaches L.

Note that the i-th pixel in the permuted pixel sequence **P'** is the $t(i)$-th pixel in the original pixel sequence **P**.

3.2. Diffusion Process

Using another initial state value x_{20} and μ_2 of LSM, we generate another chaotic sequence to diffuse the pixel values of the image **P'**. The diffusion procedure presented in the proposed scheme is composed of two rounds, which are denoted by Diffusion I and Diffusion II, respectively. In the diffusion process, a equivalent secret key sequence $\mathbf{K}=\{k(1), k(2), \cdots, k(L)\}$ is generated.

3.2.1. The First Round Diffusion

In Diffusion I, we first calculate the equivalent secret keys according to the chaotic values and values of the image pixels. Then the current pixel of image **C**' can be obtained by combining the current pixel of image **P**', the previous pixel of image **C**' and the current equivalent key of **K**. Suppose an 8-bit equivalent keys sequence $\{k(i), i=1,2,\ldots,L\}$ is available, one can compute the cipher-image pixel sequence $\{c'(i), i=1,2,\ldots,L\}$ from the image pixel sequence $\{p'(i), i=1,2,\ldots,L\}$ by using $\{k(i), i=1,2,\ldots,L\}$.

The operation steps in Diffusion I are as follows:

Step 1 Let $i\leftarrow 0$.

Step 2 Let $i\leftarrow i+1$.

Step 3 If $i=1$, then Let $c_-\leftarrow C_0$, $p_-\leftarrow p'(2)$; If $1<i<L$, then $c_-\leftarrow c'(i-1)$, $p_-\leftarrow p'(i+1)$; If $i=L$, then $c_-\leftarrow c'(i-1)$, $p_-\leftarrow K_0$.

Step 4 Let $x=x_{20}\times c_-/255, \mu=\mu_2\times p_-/255$. Iterating the LSM for one times to obtain a new x.

Step 5 Computing $k(i)$ by using current x according to the following formula (3):

$$k(i)=\mathrm{mod}(\mathrm{floor}(x\times 10^{14}),255)+1. \quad (3)$$

Step 6 To encrypt the i-th pixel by using the following formula (4):

$$c'(i)=\mathrm{mod}(p'(i)\oplus k(i)+c_-\oplus k(i),256) \quad (4)$$

Step 7 To repeat Step 2 to Step 6 until i reaches L.

3.2.2. The Second Round Diffusion

In Diffusion II, we first calculate the equivalent secret keys according to the chaotic values and values of the image pixels. Then the current pixel of image **C** can be obtained by combining the current pixel of image **C**', the previous pixel of image **C** and the current equivalent key of **K**. One can compute the cipher-image pixel sequence $\{c(i), i=1,2,\ldots,L\}$ from the image pixel sequence $\{c'(i), i=1,2,\ldots,L\}$ by using $\{k(i), i=1,2,\ldots,L\}$.

The operation steps in Diffusion II are as follows:

Step 1 Let $i \leftarrow 0$.

Step 2 Let $i \leftarrow i+1$.

Step 3 If $i=1$, then Let $c_{-} \leftarrow c'(L)$, $p_{-} \leftarrow c'(i+1)$; If $1<i<L$, then $c_{-} \leftarrow c(i-1)$, $p_{-} \leftarrow c'(i+1)$; If $i=L$, then $c_{-} \leftarrow c(i-1)$, $p_{-} \leftarrow C_0$.

Step 4 Let $x=x_{20} \times c_{-}/255$, $\mu=\mu_2 \times p_{-}/255$. Iterating the LSM for one times to obtain a new x.

Step 5 Computing $k(i)$ by using current x according to the formula (3).

Step 6 To encrypt the i-th pixel by using the following formula (5):

$$c(i) = \mathrm{mod}(c'(i) \oplus k(i) + c_{-} \oplus k(i), 256) \quad (5)$$

Step 7 To repeat Step 2 to Step 6 until i reaches L.

3.3. Decryption Algorithm

The decryption procedures are similar to those of encryption except the following modifications: permutation and diffusion are executed in reverse order. Diffusion I and Diffusion II must also be executed reversely. Eqs. (4) and (5) should be replaced with Eqs. (7) and (6), respectively. We list the key formulas according to the operation order as follows.

In the first round of decryption, decrypt the i-th pixel by using the following formula (6) (i=L,L-1,…,2,1):

$$c'(i) = \mathrm{mod}(c(i) - c_{-} \oplus k(i) + 256, 256) \oplus k(i) \quad (6)$$

In the second round of decryption, decrypt the i-th pixel by using the following formula (7)(i=L,L-1,…,2,1):

$$p'(i) = \mathrm{mod}(c'(i) - c_{-} \oplus k(i) + 256, 256) \oplus k(i) \quad (7)$$

Do reverse permutation operation on **P'**=$\{p'(i), i=1, 2,\ldots, L\}$ according to formula (8) with the permutation sequence $\{t(i)\}$. Then we can obtain the one dimensional vector **P**=$\{p(i), i=1, 2,\ldots, L\}$ of the decrypted image.

$$p(t(i))=p'(i),\ i=1, 2,\ldots, L \quad (8)$$

4. Experimental Results and Security Analysis

In our experiments, the images for testing are the 256×256 traditional images with 8-bit gray-scale. The initial secret parameters are x_{10}=0.35, x_{20}=0.77, μ_1=3.3, μ_2=3.1; N_0 =1000, C_0=225, and K_0=125.

4.1. Key Space Analysis

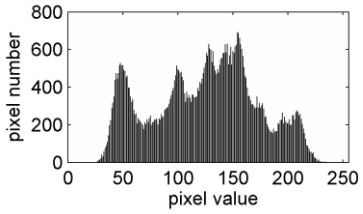
Key space size is the total number of different keys which can be used in the encryption process. In the proposed algorithm, the secret keys set **Key**=$\{x_{10}, \mu_1, x_{20}, \mu_2\}$. They are all double-precision numbers, If the computational precision of double-precision numbers is 10^{-16}, Therefore, the key space is bigger than $10^{16} \times 10^{16} \times 10^{16} \times 10^{16} = 10^{64}$, which is much larger than 2^{212}. So the encryption algorithm has a large enough key space to resist all kinds of brute force attacks.

4.2. Statistical Analysis

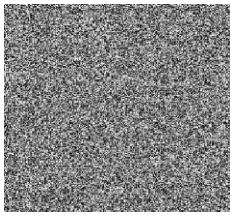
4.2.1. Histograms of Encrypted Images

In Figure 3, we give a typical example showing histograms of the plain-image and the corresponding cipher-image. As shown in Figure 3(d), all the gray-scale values of the cipher-image of ‘‘Lena’’ are distributed uniformly over the interval [0,255], which is significantly different from the original distribution shown in Figure 3(b).

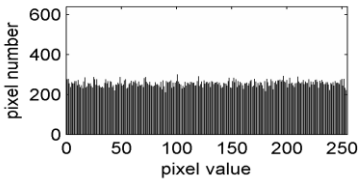
(a) Plain-image “Lena”



(b) Histogram of “Lena”



(c) The cipher-image of “Lena”



(d) Histogram of the cipher-image

Figure 3. Histograms of Plain-image “Lena” and Its Corresponding Cipher-Image

4.2.2. Correlation of Adjacent Pixels

Adjacent pixels of digital images have the intrinsic characteristic of high correlation. An effective image encryption algorithm should be able to remove this kind of relationship. To test the correlation between horizontally, vertically and diagonally adjacent pixels, we calculate the correlation coefficient in each direction by

$$\mathrm{cov}(x,y)=\frac{\frac{1}{L}\sum_{i=1}^{L}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{(\frac{1}{L}\sum_{i=1}^{L}(x_i-\overline{x})^2)\cdot(\frac{1}{L}\sum_{i=1}^{L}(y_i-\overline{y})^2)}} \tag{9}$$

where $\overline{x}=\frac{1}{L}\sum_{i=1}^{L}x_i$, $\overline{y}=\frac{1}{L}\sum_{i=1}^{L}y_i$, (x_i, y_i) is the i-th pair of adjacent pixels in the same direction and L is the total number of pixel pairs. The results of the correlation coefficients along the three directions of some images after encryption under the secret keys mentioned above are listed in Table 1. It is clear that each correlation is almost reduced to 0 after encryption. From Table 1, one can see that the encryption scheme satisfies zero co-correlation. Compared with the algorithms proposed by Ref. [14], it shows superior performance.

Table 1. Correlation Coefficients of the Ciphered Images

Correlation	Horizontal	Vertical	Diagonal
Encrypted Lena	0.000516	-0.001957	0.000543
Encrypted Lena[14]	0.032107	0.027188	0.038393
Encrypted Sailboat	0.002414	-0.001797	-0.003646
Encrypted Sailboat[14]	0.032347	0.011135	0.014651
Encrypted Pepper	0.000308	0.000080	0.002437
Encrypted Pepper[14]	0.014260	−0.00820	0.063500
Encrypted	0.001144	0.007437	0.006506

4.2.3. Information Entropy Analysis

Information entropy can offer a measure to quantify the random-looking distribution. The most commonly used information entropy is the Shannon entropy [1], which is regarded as the vital feature of randomness. For a message source with 2^n symbols s_i, its Shannon entropy, H(s), is defined as follows:

$$H(s) = -\sum_{i=0}^{2^n-1} P(s_i) \log_2[P(s_i)] \tag{10}$$

Where $P(s_i)$ is the probability of occurrence of the symbol s_i which is determined by the source. It is easy to prove that a random gray-scale image with uniformly distributed pixels over the interval [0, 255] can achieve the ideal Shannon entropy 8. Set the aforesaid secret keys, and get the cipher-image of "Len", "Baboon" and "Pepper" of the same size 256×256. The Shannon entropy of the three cipher-images are H_{Lena}=7.9975, H_{Baboon}=7.9973 and H_{Pepper}=7.9977, which are very close to the ideal value 8. Ref [16] reported the Shannon entropy of the three cipher-images are H_{Lena}=7.9973, H_{Baboon} =7.9971 and H_{Pepper} =7.9969.

4.3. Differential Attack

Generally speaking, an opponent may make a slight change (*e.g.*, modify only one pixel) of the encrypted image to observe the change in the result. In this way, we may be able to find out a meaningful relationship between the plain image and the cipher image. This is known as the differential attack. However, if the minor modification in the plain-image or secret keys generates significant and unpredictable results in the cipher-image, the differential attack will become inefficient and useless.

To give a quantitative description of the one-pixel change on the encrypted results, two common measures NPCR (number of pixels change rate) and UACI (unified average changing intensity) are used. They are defined by

$$NPCR = \frac{1}{M_1 \times M_2} \sum_{i,j} D(i, j) \times 100\% \tag{11}$$

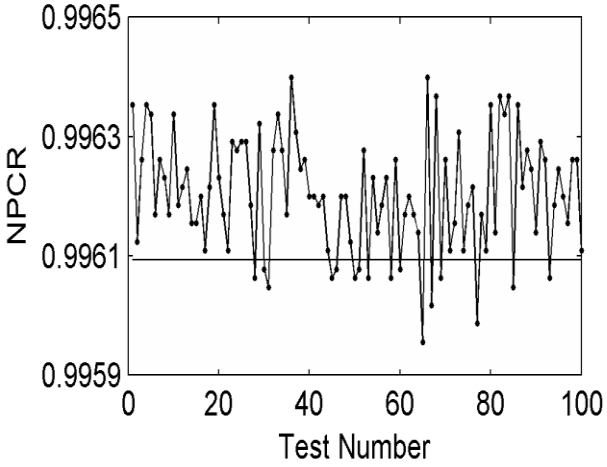
$$UACI = \frac{1}{M_1 \times M_2} \sum_{i,j} \frac{\left| c_1(i, j) - c_2(i, j) \right|}{255} \times 100\% \tag{12}$$

Where $D(i, j)$ represents the difference between $c_1(i, j)$ and $c_2(i, j)$. If $c_1(i, j)=c_2(i, j)$ then $D(i, j)=0$, otherwise $D(i, j)=1$. For an 8-bit gray image, the expected estimates are: $NPCR_E$ = 99.6094%, $UACI_E$ = 33.4635%.

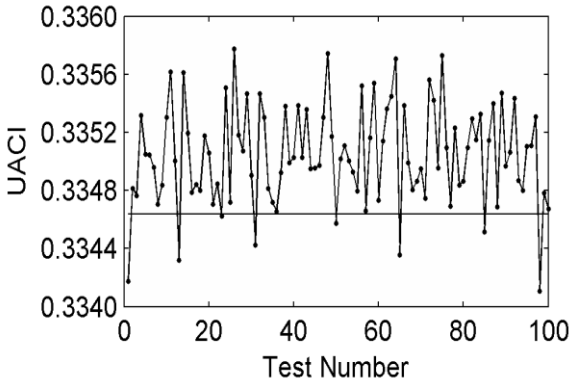
The tests of the proposed scheme are carried out as follows. Given a plain-image P_1, randomly choose a location (i, j) and obtain a plain-image P_2 by

$$p_2(i,j) = \begin{cases} p_1(i,j)+1, if \;\; p_1(i,j) < 255 \\ p_1(i,j)-1, if \;\; p_1(i,j) = 255 \end{cases} \quad (13)$$

Encrypt P_1 and P_2 and denote the cipher-images by C_1 and C_2, then one can calculate NPCR and UACI as defined above. Given the aforesaid secret keys, repeat this test 100 times, The results of NPCR and UACI are shown in Figure 4(a) and Figure 4(b), respectively. It is clear that the NPCR and UACI values remain in the vicinity of the expected values (shown by the horizontal lines), *i.e.*, the proposed image encryption technique shows extreme sensitivity to the plaintext. Ref [16] reported the mean NPCR and UACI values are 99.6041% and 33.4198%, respectively. In the proposed scheme, the mean NPCR is 99.6203% and the mean UACI is 33. 5043%.



(a) NPCR values for 100 modified plain-images peppers



(b) UACI values for 100 modified plain-images peppers

Figure 4. NPCR and UACI Values for 100 Modified Plain-images Peppers

A good encryption algorithm should also be sensitive to the secret keys. For the LSM system, the sensitivity to the keys is tested. The NPCR and UACI between two encrypted images with keys (x_{10}, μ_1, x_{20}, μ_2) and slightly varied keys (only one of the four parameters has varied 10^{-10}) are shown in Table 2. It shows that the NPCR and UACI are all close to the ideal values. Therefore, the encryption algorithm is very sensitive to the secret keys.

Table 2. NPCR and UACI Values with Slightly Varied Keys

modified keys	NPCR (%)	UACI (%)
$\Delta x_{10}=10^{-10}$	99.6552	33.3231
$\Delta x_{20}=10^{-10}$	99.2767	33.4837
$\Delta \mu_1=10^{-10}$	99.5987	33.3037
$\Delta \mu_2=10^{-10}$	99.2340	33.4072

5. Conclusion

This paper proposes a symmetric image encryption scheme by using the new LSM chaotic system and the improved permutation-diffusion structure. As the equivalent permutation sequence and secret diffusion key sequence are all related to the plain-image, one can develop different permutation and diffusion sequences for different plain-images, which makes the scheme immune to known/chosen plaintext attack. Experimental tests demonstrate that the scheme possesses large key space, uniform distribution of cipher-images and high sensitivity to plain image and keys. So the proposed scheme has a good ability to resist brute-force attacks, statistical analysis attacks and differential attacks. With high-level security, it can be used in secure image communications.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No. 61073187), and key Project Supported by Scientific Research Fund of Guangxi Provincial Education Department (Grant No. 201202ZD080).

References

[1] C. E. Shannon, “Communication theory of secrecy systems”, Bell Syst Tech J, vol. 28, no. 4, **(1949),** pp. 656–715.
[2] J. Fridrich, “Symmetric ciphers based on two-dimensional chaotic maps”, International Journal of Bifurcation and Chaos, vol. 8, no. 6, **(1998),** pp. 1259-1284.
[3] G. R. Chen, Y. B. Mao and C.K. Chui, “A symmetric image encryption scheme based on 3D chaotic cat maps”, Chaos, Solitons & Fractals, vol. 21, no. 3, **(2004),** pp. 749-761.
[4] C. X. Zhu, “A novel image encryption scheme based on improved hyperchaotic sequences”, Optics Communications, vol. 285, no. 1, **(2012),** pp. 29-37.
[5] X. J. Tong and M. G. Cui. “Image encryption with compound chaotic sequence cipher shifting dynamically”, Image and Vision Computing, vol. 26, no. 6, **(2002),** pp. 843-850.
[6] H. J. Liu and X. Y. Wang, “color image encryption based on one-time keys and robust chaotic maps”, Computers & Mathematics with Applications, vol. 59, no. 10, **(2010),** pp. 3320-3327.
[7] Z. L. Zhu, W. Zhang, K. W. Wong, *et al.*, “A chaos-based symmetric image encryption scheme using a bit-level permutation”, Information Sciences, vol. 181, no. 6, **(2011),** pp. 1171-1186.
[8] Z. H. Guan, F. J. Huang and W. J. Guan, “Chaos-based image encryption algorithm”, Physics Letters A, vol. 346, no. 1-3, **(2005),** pp. 153-157.
[9] D. Xiao, X. F. Liao and P. C. Wei, Chaos, “Analysis and improvement of a chaos-based image encryption algorithm”, Chaos, Solitons & Fractals, vol. 40, no. 5, **(2005),** pp. 2191-2199.
[10] G. J. Zhang and Q. LIU, “A novel image encryption method based on total shuffling scheme”, Optics Communications, vol. 284, no. 12, **(2011),** pp. 2775-2780.
[11] Y. Zhou, L. Bao and C. L. P. Chen, “A new 1D chaotic system for image encryption”, Signal Processing, vol. 97, **(2014),** pp. 172-182.
[12] G. Ye, “Image scrambling encryption algorithm of pixel bit based on chaos map”, Pattern Recognition Letters, vol. 31, no. 5, **(2010),** pp. 347-354.
[13] X. Y. Wang, L. Teng and X. Qin, “A novel colour image encryption algorithm based on chaos”, Signal Processing, vol. 92, no. 4, **(2012),** pp. 1101-1108.

[14] X. Y. Wang and C. Q. Jin, "Image Encryption Using Game of Life Permutation and LSM Chaotic System", Optics Communications, vol. 285, no. 4, **(2012),** pp. 412–417.
[15] X. Y. Wang and D. H. Xu, "A novel image encryption scheme based on Brownian motion and PWLCM chaotic system", Nonlinear Dynamics, vol. 75, no. 1-2, **(2014),** pp. 345-353.
[16] L. Y. Zhang, X. B. Hu, Y. S. Liu, K.-W. Wong and J. Gan, "A chaotic image encryption scheme owning temp-value feedback", Communications in Nonlinear Science and Numerical Simulation, vol. 19, no. 10, **(2014),** pp. 3653-3659.
[17] C. Li and K. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks", Signal Process, vol. 91, no. 4, **(2011),** pp. 949–54.
[18] S. Li, C. Li, G. Chen, N. G. Bourbakis and K.-T. Lo, "A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks", Signal Processing: Image Communication, vol. 23, no. 3, **(2008),** pp. 212–23.
[19] S. B. He, K. H. Sun and C. X. Zhu, "Complexity analyses of multi-wing chaotic systems", Chinese Physics B, vol. 22, no. 5, **(2013),** pp. 050506-1~6.
[20] K. H. Sun, S. B. He, C. X. Zhu and Y. He, "Analysis of chaotic complexity characteristics based on C_0 algorithm", Acta Electronica Sinica, vol. 41, no. 9, **(2013),** pp. 1765-1771.

Authors

Congxu Zhu, he received the Ph.D. degree in computer applied technology from Central South University, China in 2006. Currently he is a professor at Central South University, China. His research interests include chaos theory and its applications in information security, chaos-based cryptography, image processing, multimedia and network security.

YuPing Hu, he received his B. S. degree in celestial survey from Chinese Academy of Science, China in 1996 and his Ph.d. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 2005. He is a professor in the Guangdong University of Finance & Economics since 2006. His current research interests include digital watermarking, image processing, multimedia and network security.

Xinran Zhou, he received the Ph. D. degree in control theory and control engineering from Hunan University, China in 2013. Currently he is a lecturer at Central South University, China. His research interests include pattern recognition and intelligent system, intelligent information processing, multimedia and network security.

