Postprint: An Image Encryption Method Based on Composite Chaotic Sequences

Authors: Zhang Xiaobo, Peng Jinye, Xi Min

Date: 2018-10-11T00:00:00+00:00

Abstract

To address the limitations of small key space and low security in low-dimensional chaotic systems, an image encryption method is proposed that forms a composite chaotic sequence by using Sine chaos to alter the ordering of uniformly distributed Logistic chaotic sequences. First, a Logistic chaotic sequence following a uniform distribution is generated, and a Sine chaotic sequence is used to rearrange the repeated portions after integer conversion of this sequence; this composite chaotic sequence with non-repeating values is then employed for pixel position scrambling. Subsequently, since position scrambling alone cannot alter the grayscale statistical histogram characteristics of the image, Sine chaos is used to rearrange the entire Logistic chaotic sequence to form a composite chaotic sequence, which is then utilized for pixel diffusion to complete the image encryption. The security of the method is evaluated in terms of key space, key sensitivity, differential analysis, statistical histogram, adjacent pixel correlation, and information entropy. Experimental results demonstrate that the method possesses a large key space and high sensitivity, and can effectively resist brute-force analysis, differential analysis, and statistical analysis.

Full Text

Abstract

Low-dimensional chaotic systems suffer from small key spaces and insufficient encryption security. This paper proposes an image encryption method that employs composite chaotic sequences formed by using Sine chaos to alter the ordering of uniformly distributed Logistic chaos. First, a uniformly distributed Logistic chaotic sequence is generated, and the Sine chaotic sequence is used to rearrange the repeated portions after integer conversion, creating a composite chaotic sequence without duplicate values for pixel position scrambling. Since position scrambling alone cannot alter the grayscale histogram characteristics of an image, the Sine chaotic sequence is then used to reorder the entire Logistic

chaotic sequence to form a composite sequence for pixel diffusion, completing the image encryption. The security of the method is evaluated in terms of key space, key sensitivity, differential analysis, statistical histograms, adjacent pixel correlation, and information entropy. Experimental results demonstrate that the proposed method offers a large key space and high sensitivity, effectively resisting exhaustive, differential, and statistical analyses.

Keywords: Image encryption; uniform distribution; Logistic chaos; Sine chaos

0 Introduction

With the increasing maturity of internet technology and the development of information technology, more information is being digitized and transmitted over networks, including text, images, and video. Due to their intuitive and vivid characteristics, images are widely used for information dissemination and exchange. Ensuring the security and confidentiality of transmitted images in complex network environments is of great significance. Traditional encryption methods such as IDEA, AES, and DES, primarily designed for text data, are not well-suited for image data characterized by large data volumes and strong correlations between pixels.

Chaotic systems generate continuous signals that, after discretization, form chaotic sequences with unpredictability and non-periodicity, offering distinct advantages for image encryption. Research on image encryption based on chaotic systems has become a hot topic. Literature [4] proposed using Logistic mapping and two-dimensional Henon mapping to complete encryption through two steps: image position scrambling and pixel grayscale value encryption. Literature [5] employed Logistic mapping for both position scrambling and grayscale value diffusion. Literature [6] performed position scrambling followed immediately by diffusion for each pixel. These methods, compared to single position scrambling, alter both pixel positions and grayscale value distributions, increasing the difficulty of cryptanalysis. However, the statistical characteristics of Logistic chaos used in these methods do not follow a uniform distribution, resulting in insufficient ciphertext concealment.

The security of chaotic encryption largely depends on the distribution, randomness, and complexity of chaotic sequences [7], making it beneficial to modify basic chaotic sequences. Literature [8] addressed the limitation of the one-dimensional Logistic mapping having only two variable parameters by proposing a method based on three-dimensional interleaved coding Logistic mapping to generate chaotic sequences. Literature [9] presented a color image encryption method combining Logistic mapping and double random phase encoding. Literature [10] proposed a Logistic-Chua composite chaotic mapping method composed of Logistic mapping and Chua' s circuit. Literature [11] used XOR, modulo, and feedback operations between pairs of chaotic mappings such as Logistic, Tent, and Sine to form new composite chaos. Literature [12] uniformized the Logistic chaotic sequence and implemented a Logistic random permutation

based on position interchange (LRPRI), which further increased the disorder of Logistic through position exchange on uniformly distributed sequences, though the security of using a single chaotic sequence as the key sequence needs improvement.

Building on literature [11,12], this paper proposes a composite chaotic sequence generation method. First, a uniformly distributed Logistic chaotic sequence is generated, and then the numerical order of the Sine chaotic sequence is used to rearrange the element positions of the Logistic sequence to form a composite chaotic sequence. This approach changes the ordering of chaotic sequences through different types of chaotic systems, maintaining the uniform distribution characteristics of the original chaotic sequence while enhancing the disorder of the key sequence through reordering.

The encryption process consists of two steps: position scrambling and pixel diffusion. In position scrambling, to avoid duplicate values caused by rounding errors when converting chaotic sequence values to integers, we propose a composite chaotic sequence generation method without duplicate values using the Sine sequence to reorder portions of the Logistic sequence. In pixel diffusion, to further increase the disorder of the Logistic sequence, the Sine sequence is used to reorder the entire Logistic sequence to generate a composite chaotic sequence for diffusion. Experiments demonstrate that the proposed method effectively resists exhaustive, statistical, and differential attacks.

1 Chaotic Sequences

Image encryption based on chaotic sequences typically involves operations such as XOR and cyclic shifting between chaotic sequences and original image information to transform the original image into a state resembling random noise, achieving encryption. The chaotic sequences used in this paper are described below.

1.1 Basic Logistic Chaotic Sequence

Chaos in dynamical systems refers to unpredictable, random-like motion exhibited by deterministic systems due to sensitivity to initial conditions. Logistic chaotic mapping is commonly used for image encryption in image processing. The Logistic chaotic mapping is a dynamical system with the equation:

$$x_{i+1} = \mu x_i(1 - x_i), \quad x_i \in V, i = 0, 1, 2, ...$$

where x_i is called the state, x_0 is the initial value, and μ is the Lyapunov exponent of the chaotic system with a value range of $(0, 4]$. When the Lyapunov exponent μ is in the interval $(3.5699456, 4.0]$, the Logistic mapping exhibits chaotic behavior, and its iteratively generated sequence values are in a random distribution state with a system state value domain of $(0, 1)$. The basic Logistic

mapping generates non-uniformly distributed sequences, which poses certain security risks when directly used for image encryption.

1.2 Uniformly Distributed Chaotic Sequence

There have been studies on converting chaotic sequences into sequences with uniform distribution statistical characteristics [12-14]. This paper adopts the sequence mapping method for conversion [12]. For the basic Logistic chaotic sequence x_i, it is mapped and converted into a uniformly distributed chaotic sequence x_i^2 according to equation (2) as follows:

$$x_i^2 = \frac{1}{\pi} \arcsin(2x_i - 1) + \frac{1}{2}$$

After conversion, the sequence x_i^2 becomes a random variable uniformly distributed on the interval $(0, 1)$. With $\mu = 3.997$ and $x_0 = 0.512$, iterating 30,000 times produces a Logistic sequence whose value distribution histogram is shown in Figure 1: see original paper. The statistical histogram of the uniformly distributed Logistic sequence after conversion according to equation (2) is shown in Figure 1: see original paper. From the statistical histogram of sequence value frequencies in Figure 1: see original paper, it can be seen that after mapping conversion, the probability of sequence values appearing in the $(0, 1)$ interval for the Logistic sequence is uniform.

1.3 Sine Chaotic Sequence

The Sine mapping used to construct the composite chaotic sequence is expressed as:

$$x_{n+1} = a \sin(\pi x_n), \quad 0 < x_n \leq 1$$

where the system parameter a has a value range of $(0, 4]$. Since the value domain interval of the Sine function is $[-1, 1]$, it is computationally convenient when reordering other sequences based on its sequence value magnitudes. Therefore, Sine chaos is selected to change the ordering of the Logistic chaotic sequence to form a composite chaotic sequence.

2 Image Pixel Position Scrambling

Pixel position scrambling aims to obscure the relationship between plaintext, key, and ciphertext from the spatial position of the image. The method uses a composite chaotic sequence without duplicate values as the scrambling key sequence to change pixel positions, achieving position scrambling.

2.1 Non-Repeating Scrambling Key Sequence

This paper proposes a method for generating non-repeating composite chaotic sequences based on uniformly distributed Logistic sequences. The method uses a Sine-type chaotic sequence to change the arrangement of the uniformly distributed Logistic sequence, forming a new composite chaotic sequence. Considering that directly using this composite chaotic sequence for image position scrambling requires mapping Logistic sequence values in the range $(0, 1)$ to image pixel coordinates, the common approach is to multiply sequence values by the image size and then take integers. Floating-point integer conversion through rounding produces a considerable number of identical integers. When using sequences with identical integers to change plaintext image pixel positions, the corresponding pixel positions will not be moved, failing to achieve the scrambling purpose.

To solve this problem, we propose a non-repeating scrambling key sequence generation method that uses the Sine chaotic sequence to replace duplicate values in the integerized Logistic sequence, generating a composite chaotic sequence for encryption. The steps are as follows:

a) For an image of size $M \times N$, generate a set of Logistic sequences of length L, and convert them to uniform Logistic sequences using equation (2). To ensure chaotic properties, 截取从 200 点之后长度为 M×N 的部分，记为序列 m1.

b) Since m1 values are floating-point numbers in the range $(0, 1)$, to correspond with the pixel positions of the image to be scrambled, multiply each element value in m1 by $(M \times N - 1)$, then add 1 to change each value in the sequence to floating-point numbers between $(1, M \times N)$, denoted as sequence m2.

c) Convert the values in sequence m2 to integers using rounding, which produces a considerable number of duplicate values in the sequence. Keep only the value at the position where the duplicate number first appears, set subsequent occurrences of the sequence value to 0, and record the total number of duplicate values S and their positions in m2.

d) Generate a chaotic sequence of length S using the Sine mapping, with sequence values in $[-1, 1]$.

e) Sort the Sine chaotic sequence values in ascending order and record the position of each value in the original sequence using a one-dimensional array R.

f) Insert the numbers between $[1, M \times N]$ that do not appear in m2 into the sequence positions set to 0 in step 3 according to the order of array R.

Through the above steps, a sequence m3 of length $M \times N$ with a value range of $[1, M \times N]$ and mutually different integer values is obtained, using m3 as the position scrambling key sequence.

Taking an image size of 256×256 as an example for scrambling key sequence generation. The key sequence parameters are $[\mu, x_1(0), a, x_2(0)]$, where μ is the Logistic chaotic Lyapunov exponent and $x_1(0)$ is the initial value; a is the Sine function coefficient and $x_2(0)$ is the initial value, taken as $[3.9997, 0.512, 4.0, 0.88]$. First, a uniformly distributed Logistic sequence of length 65536 is generated, with each sequence value multiplied by the image size $M \times N$ and converted to a 32-bit unsigned integer sequence. At this point, the sequence produces 24,400 duplicate values due to integer rounding. Subsequently, the values and positions of these duplicates in the Logistic sequence are recorded. Finally, a Sine chaotic sequence of length 24,400 is generated. According to the ascending order of Sine sequence values, the 24,400 numbers not appearing in the previous 1~65536 are refilled into these duplicate positions, producing an integer sequence of the same size as the image with non-repeating values from 1 to 65536. The sequence arrangement order is determined jointly by the Logistic and Sine chaotic sequences, as shown in [Figure 2: see original paper].

In [Figure 2: see original paper], the sequence length is consistent with the image size at 65536, with sequence values from 1-65536, each appearing only once.

2.2 Image Pixel Position Scrambling

Image pixel position scrambling disrupts the pixel positions of the original image through the key sequence. After generating the non-repeating scrambling key, the pixel position scrambling steps are simplified to the following four steps:

a) Generate a non-repeating integer key sequence using Logistic and Sine chaotic sequences according to the method in section 2.1.

b) Convert the image into a one-dimensional sequence.

c) Move each element in the one-dimensional sequence sequentially to the target position indicated by the corresponding value in the generated integer key sequence.

d) Convert the one-dimensional sequence back to two-dimensional, completing the pixel position scrambling of the image.

3 Image Pixel Diffusion

Image pixel scrambling visually obscures the original image content by moving pixel positions, but the grayscale values of pixels remain unchanged. The grayscale statistical histogram of the encrypted image is identical to that of the original image, posing security risks. It is necessary to hide the statistical histogram characteristics of the image through pixel diffusion, i.e., to hide the information of any pixel point in other ciphertext pixel points without changing pixel positions.

3.1 Diffusion Key Sequence Generation

The diffusion key sequence is generated using a composite chaotic sequence method similar to the scrambling key. To further increase the disorder of the key sequence, while the scrambling sequence uses the Sine sequence to change the ordering of only the duplicate values in portions of the Logistic sequence, the diffusion sequence uses the Sine sequence to change the ordering of the entire Logistic sequence. The steps are as follows:

a) Generate a one-dimensional uniform Logistic chaotic sequence of the same length as the image size $M \times N$.

b) Generate a Sine chaotic sequence of length $M \times N$ with values in $[-1, 1]$.

c) Sort the elements of the Sine chaotic sequence in ascending order and record the position of each element in the original Sine sequence using a one-dimensional array A.

d) Reorder the Logistic chaotic sequence using the Sine chaotic sequence, i.e., rearrange the Logistic sequence according to the values of one-dimensional array A to obtain the composite chaotic sequence S for pixel diffusion.

3.2 Pixel Diffusion

After generating the diffusion key sequence, bidirectional cyclic XOR operations are used for pixel diffusion, i.e., performing forward and reverse cyclic XOR operations twice on each pixel of the image. This diffuses the information of a pixel point in the plaintext image into all ciphertext pixels [15].

The pixel diffusion process: First, convert a grayscale image of size $M \times N$ into a one-dimensional vector P of length $M \times N$, whose values correspond one-to-one with the key sequence S of length $M \times N$ generated in section 3.1. Then perform forward and reverse XOR operations sequentially for two cycles to obtain the ciphertext C represented as a one-dimensional vector. The forward cyclic XOR operation (i from 1 to MN) is shown in equation (4), with the corresponding decryption inverse operation shown in equation (5):

$$C_i = P_i \oplus S_i \oplus C_{i-1}$$

$$P_i = C_i \oplus S_i \oplus C_{i-1}$$

As shown in equation (4), after forward cycling, the information of pixel point P_1 can be diffused into all ciphertext pixel information, but the information of pixel point P_{MN} can only be diffused into C_{MN}, i.e., the information of plaintext pixel point P_{MN} can only be diffused into C_{MN}, resulting in poor diffusion effect. Therefore, a reverse cycle (i from MN to 1) is needed once, i.e., according to equation (6), with the corresponding decryption inverse operation shown in equation (7):

$$C_i = P_i \oplus S_i \oplus C_{i+1}$$

$$P_i = C_i \oplus S_i \oplus C_{i+1}$$

4 Encryption and Decryption Process

When a chaotic system consists of two or more chaotic sequences, its nonlinear behavior becomes more complex and unpredictable [16]. The key sequence in this paper is composed of uniformly distributed Logistic chaos and Sine chaos. The key sequence generation method is described in section 3, with parameters $[\mu, x_1(0), a, x_2(0)]$, where μ is the Logistic chaotic Lyapunov exponent and $x_1(0)$ is the initial value; a is the Sine function coefficient and $x_2(0)$ is the initial value. The encryption and decryption processes are as follows:

4.1 Encryption Process

The encryption process includes two steps: pixel position scrambling and grayscale value diffusion. First, generate a uniformly distributed Logistic chaotic sequence and convert it to integers. Use the Sine chaotic sequence to change the ordering of portions of this sequence to produce a non-repeating chaotic sequence, which is used to scramble the pixel positions of the image. Then, using the same key parameters, generate a uniformly distributed Logistic chaotic sequence and use the Sine chaotic sequence to change the ordering of the entire Logistic sequence. This sequence is used with the scrambled image for forward and reverse diffusion operations to complete pixel grayscale value diffusion, achieving image encryption.

4.2 Decryption Process

The image decryption process is the inverse of encryption, with decryption key parameters identical to encryption keys. The decryption process first performs pixel grayscale value diffusion decryption, followed by pixel position scrambling decryption, to recover the image from ciphertext.

5 Experiments and Results

The Cameraman, Lena, and Peppers grayscale images were selected as test images for encryption and decryption experiments. The system parameters and initial values of Logistic and Sine chaos were used as keys, with values $[3.9997, 0.512, 4.0, 0.88]$. The experiments were conducted on an i5 CPU using MATLAB 2014. The average encryption time was 0.153 seconds, and the average decryption time was 0.151 seconds. The visual effects are shown in [Figure 3: see original paper].

In [Figure 3: see original paper], column (a) shows from top to bottom: Cameraman plaintext image, grayscale histogram and amplitude spectrum; Lena plaintext image, grayscale histogram and amplitude spectrum; Peppers plaintext image, grayscale histogram and amplitude spectrum. Column (b) shows from top to bottom: Cameraman ciphertext image, grayscale histogram and amplitude spectrum; Lena ciphertext image, grayscale histogram and amplitude spectrum; Peppers ciphertext image, grayscale histogram and amplitude spectrum. Column (c) shows from top to bottom: decrypted Cameraman image, grayscale histogram and amplitude spectrum; decrypted Lena image, grayscale histogram and amplitude spectrum; decrypted Peppers image, grayscale histogram and amplitude spectrum.

From the plaintext, ciphertext, and decrypted images in [Figure 3: see original paper], it can be seen that the grayscale statistical histogram of the ciphertext image is essentially uniformly distributed, and the amplitude spectrum of ciphertext image grayscale values is also very flat, i.e., the probability of ciphertext image pixel values appearing in the grayscale range [0,255] is almost equal.

6 Security Analysis

6.1 Key Space Analysis

Key space is an important indicator for measuring the security of a cryptosystem. The larger the key space, the stronger the ability to resist exhaustive attacks. This paper uses the Logistic system parameter μ, initial value $x_1(0)$, and Sine chaotic mapping parameter a, $x_2(0)$ as keys. In a 32-bit computer, according to the IEEE 754 standard, the length for representing double-precision floating-point numbers is 64 bits, so the key space is $2^{64} \times 2^{64} \times 2^{64} \times 2^{64} = 2^{256} \approx 10^{77}$. From a security perspective, a key space $\geq 2^{100} \approx 10^{30}$ can meet a high security level, so the key space of this algorithm is secure against exhaustive attacks.

6.2 Key Sensitivity Analysis

Key sensitivity refers to the ability to produce completely different ciphertext from the original ciphertext when a tiny change is made to the key during encryption (e.g., $\Delta\mu = 10^{-10}$ or $\Delta x_1(0) = 10^{-10}$). Similarly, during decryption, a tiny change to the original decryption key can produce completely different decryption results from the same ciphertext [17][18].

To evaluate key sensitivity, a tiny change is made to a key, and the change rate of the corresponding ciphertext image obtained through the encryption algorithm is calculated. The Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are commonly used for measurement, defined as follows:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%$$

$$UACI = \frac{1}{255}\frac{\sum_{i,j}|I_1(i,j) - I_2(i,j)|}{M \times N} \times 100\%$$

where I_1 is the image encrypted with the original key, I_2 is the image encrypted with the slightly changed key; $D(i,j)$ represents the number of different pixels between images I_1 and I_2, taking value 1 when $I_1(i,j) \neq I_2(i,j)$ and 0 otherwise; M, N are the length and width of images I_1 and I_2 respectively. The test image Cameraman is an 8-bit grayscale image with theoretical values: NPCR = 99.6094%, UACI = 33.4635%.

In the experiment, a tiny change was made to the encryption key for the Cameraman image during encryption, increasing the initial value of the chaotic system by 10^{-10} each time (denoted as Δ in), and measuring the NPCR and UACI values between the ciphertext generated with the slightly changed key and the original ciphertext. The results are shown in .

TABLE:1 shows that the measured NPCR and UACI values are very close to the ideal values, indicating that when the key undergoes a tiny change, more than 99% of pixels in the ciphertext image will change, making it difficult for attackers to analyze using exhaustive methods. The decryption process of this paper is symmetric to the encryption process, so the same conclusion can be drawn for decryption keys, i.e., the key sensitivity of the proposed method is very strong.

6.3 Differential Attack Analysis

Differential attack is a chosen-plaintext attack where attackers analyze the differences between corresponding ciphertexts after making tiny changes to the plaintext encrypted by the same system. The ability of an encryption system to resist differential attacks can be measured by two indicators: NPCR and UACI [10,19]. Experiments used Cameraman, Lena, and Peppers grayscale images with the following steps:

a) For plaintext image I, use the encryption system to obtain the corresponding ciphertext image E_1.

b) Randomly select a pixel point from image I, change its grayscale value by 1, and obtain the ciphertext image E_2 using the same key and encryption system.

c) Calculate NPCR and UACI for ciphertexts E_1 and E_2.

d) Repeat steps b) and c) 100 times to obtain average NPCR and UACI values.

TABLE:2 shows the average NPCR and UACI values obtained from the experiments, which are close to the theoretical values, indicating that the encryption algorithm has good resistance to differential attacks.

TABLE:2 Average NPCR and UACI values for differential attack analysis

Image	NPCR	UACI
Cameraman	99.61%	33.42%
Lena	99.62%	33.45%
Peppers	99.60%	33.44%

6.4 Statistical Analysis

Statistical analysis of encryption algorithms aims to measure their ability to resist statistical attacks in terms of scrambling and diffusion performance. This is achieved through analysis of grayscale statistical histograms, adjacent pixel correlation, and information entropy.

6.4.1 Grayscale Statistical Histogram The grayscale statistical histogram of an image characterizes the distribution of image pixels by counting the number of occurrences of each grayscale level. The grayscale statistical histograms of plaintext and ciphertext images before and after encryption are shown in [Figure 3: see original paper]. The figure shows that the ciphertext image histogram is uniformly distributed, and attackers can hardly derive statistical characteristics from the histogram.

6.4.2 Adjacent Pixel Correlation Analysis Images are composed of pixels, and adjacent pixels have certain correlations due to color and brightness transitions. Grayscale images can be viewed as matrices composed of different pixel grayscale values. Since there is correlation between adjacent pixels, the grayscale value differences between many adjacent pixels are small. To improve the confidentiality of encrypted images, the correlation between adjacent pixels in ciphertext images must be reduced [20]. The correlation coefficient ρ_{xy} between adjacent pixels is defined as follows [20]:

$$\rho_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

where x and y represent the grayscale values of two adjacent pixels in the image, $cov(x, y)$ is the covariance of grayscale values x and y, and $D(x)$ and $D(y)$ are the variances of x and y respectively, calculated as:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$$

From the above definitions, the stronger the correlation between adjacent pixels, the larger the correlation coefficient ρ_{xy}.

In the Cameraman plaintext and ciphertext images, 10% of pixel points were randomly selected, and the grayscale distribution of adjacent pixels in horizontal, vertical, and diagonal directions was taken, as shown in [Figure 4: see original paper].

In [Figure 4: see original paper], the grayscale values of adjacent pixels in the plaintext in three directions are almost concentrated and distributed along a straight line, indicating small differences in grayscale values between each adjacent pixel pair and strong correlation. In contrast, the grayscale values of adjacent pixel pairs in the ciphertext are evenly distributed across the entire grayscale range, approaching no correlation. Therefore, the proposed algorithm can destroy the correlation between adjacent pixels, making the distribution of ciphertext pixel grayscale values in the image plane approach randomness, which is difficult for attackers to crack using statistical attack methods.

TABLE:3 compares the average correlation coefficients between adjacent pixels of images using the proposed method. The table shows that the average correlation coefficients between adjacent pixels of ciphertext images are significantly reduced compared to plaintext images, approaching 0.

TABLE:3 Average correlation coefficients between adjacent pixels in plaintext and ciphertext

Image	Plaintext	Ciphertext
Cameraman	0.9214	0.0032
Lena	0.9236	0.0028
Peppers	0.9187	0.0035

6.4.3 Information Entropy Information entropy is a physical quantity that measures information uncertainty, calculated as:

$$H(X) = -\sum_{i=1}^{L} P(x_i)\log_2 P(x_i)$$

where $P(x_i)$ is the probability of signal x_i occurring, and L is the number of bits per signal. For images, L represents the number of grayscale levels. The greater the information uncertainty, the larger the entropy value.

The experiment was conducted on the Cameraman image, which is an $L = 8$ grayscale image with 256 grayscale levels. The maximum information entropy value is 8. The information entropy of the Cameraman image encrypted by the proposed method is calculated as 7.9895, which is close to the maximum information entropy value, indicating that the encrypted image has high information uncertainty and can resist entropy analysis.

7 Conclusion

This paper proposes a new composite chaotic sequence generation method and implements image encryption through dual encryption steps of pixel position scrambling and grayscale value diffusion. The main features of the method are:

a) The method of changing the ordering of uniformly distributed Logistic chaotic sequences based on the element ordering of Sine chaotic sequences is computationally simple.

b) In the position scrambling process, the Sine chaotic sequence is used to reorder the duplicate elements in portions of the uniformly distributed Logistic sequence, generating a key sequence without duplicate values that corresponds one-to-one with the pixel positions of the image to be scrambled, simplifying subsequent scrambling steps.

c) In pixel value diffusion, the Sine chaotic sequence is used to reorder the entire uniformly distributed Logistic sequence, which further increases the disorder of the sequence compared to partial reordering, and completes pixel diffusion through forward and reverse cyclic XOR operations.

The scrambling and diffusion processes of the proposed encryption scheme are performed in the spatial domain, preserving the original image information completely. Experimental verification demonstrates that the method can effectively resist exhaustive analysis, differential analysis, and statistical analysis attacks.

References

[1] Pareek N K, Patidar V, Sud K K. Image encryption using chaotic logistic map [J]. Image and Vision Computing, 2006, 24 (9): 926-934.

[2] Li Chengqing, Tao Xie, Qi Liu, Cheng Ge. Cryptanalyzing image encryption using chaotic logistic map [J]. Nonlinear Dynamics, 2014, 78 (2): 1545-1551.

[3] Gao Tiegang, Chen Zengqiang. A new image encryption algorithm based on hyper-chaos [J]. Physics Letters A, 2008, 372 (4): 394-400.

[4] Zhang Xuefeng, Fan Jiulun. Extended image encryption algorithm based on chaos system [J]. Application Research of Computers, 2007, 24 (4): 184-186.

[5] Hu Chunqiang, Deng Shaojiang, Qin Mingfu, et al. Image encryption algorithm based on logistic and standard map [J]. Computer Science, 2010, 37 (12): 57-59.

[6] Shu Yonglu, Zhang Yushu, Xiao Di, et al. Image encryption algorithm based on the synchronization of permutation and diffusion [J]. Journal of Lanzhou University: Natural Sciences, 2012, 48 (2): 113-116.

[7] Zhu Congxu, Hu Yuping, Sun Kehui. New image encryption algorithm based on hyperchaotic system and ciphertext diffusion in crisscross pattern [J]. Journal of Electronics & Information Technology, 2012, 34 (7): 1735-1743.

[8] Ye Guodong, Huang Xiaoling. An efficient symmetric image encryption algorithm based on an intertwining logistic map [J]. Neurocomputing, 2017, 251: 45-53.

[9] Huang Huiqing, Yang Shouzhi. Colour image encryption based on logistic mapping and double random-phase encoding [J]. Iet Image Processing, 2017, 11 (4): 211-216.

[10] Slimane N B, Bouallegue K, Machhout M. Designing a multi-scroll chaotic system by operating Logistic map with fractal process [J]. Nonlinear Dynamics, 2017, 88 (3): 1655-1675.

[11] Zhou Yicong, Bao Long, Chen C. L. Philip. A new 1D chaotic system for image encryption [J]. Signal Processing, 2014, 97: 172-182.

[12] Cao Guanghui, Hu Kai, Tong Wei. Image scrambling based on Logistic uniform distribution [J]. Acta Physica Sinica, 2011, 60 (11): 133-140.

[13] Sheng Liyuan, Xiao Yanyu, Sheng Zhe. A universal algorithm for transforming chaotic sequences into uniform pseudo-random sequences [J]. Acta Physica Sinica, 2008, 57 (7): 4007-4012.

[14] Li Peiyue, Shi Junxia, Guo Jialiang, et al. Improvement of a Universal Algorithm for Uniformization of Chaotic Pseudo-Random Sequences [J]. Acta Electronica Sinica, 2015, 43 (4): 753-759.

[15] Zhang Yong. Chaotic Digital Image Cryptosystem [M]. Beijing: Tsinghua University Press, 2016: 59-60.

[16] Jiang Junli, Zhang Xuefeng. Color image encryption method based on chaotic systems [J]. Application Research of Computers, 2014, 31 (10): 3131-3136.

[17] Luo Yuling, Du Minghui. Image encryption algorithm based on quantum logistic map in wavelet domain [J]. Journal of South China University of Technology: Natural Science Edition, 2013, 41 (6): 53-62.

[18] Behnia S, Akhshani A, Ahadpour S, et al. A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps [J]. Physics Letters A, 2007, 366: 391-396.

[19] Xu Lu, Li Zhi, Li Jian, Hua Wei. A novel bit-level image encryption algorithm based on chaotic maps [J]. Optics and Lasers in Engineering, 2016, 78 (21): 17-25.

[20] Mohamad Javad Rostam, Abbas Shahba, Saeid Saryazdi, et al. A novel parallel image encryption with chaotic windows based on logistic map [J]. Computers and Electrical Engineering, 2017, 62: 384-400.

Note: Figure translations are in progress. See original paper for figures.