PROCEEDINGS OF SCIENCE

A New Image Encryption Algorithm by Combining Two Chaotic Maps

Congxu Zhu[1, 2]
School of Information Science and Engineering, Central South University
Changsha, 410083, China
Guangxi Colleges and Universities Key Laboratory of Complex System Optimization and Big Data Processing, Yulin Normal University
Yulin, 537000, China.
E-mail: zhucx@csu.edu.cn

Shuai Li
School of Information Science and Engineering, Central South University
Changsha, 410083, China
E-mail: lishuaizx@sina.com

This treatise proposes a new algorithm of encrypting images by combining Sine map with generalized Arnold transformation. The algorithm adds a shift transformation link to the common permutation-diffusion structure. Firstly, the plain-image is permutated in pixel positions by using the generalized Arnold transformation. Secondly, the permutated image is diffused by using the Sine map. Then, the diffused image is performed through the shifting process, which can avoid multiple encryption and shorten the encryption time. The results of analysis and experimental tests for the proposed algorithm have been given in detail, which showed that our new algorithm is highly secure. In conclusion, it has great application potential in Internet-based image secure communication.

ISCC2017
16-17 December 2017
Guangzhou, China

[1]Speaker

[2]This study is supported by the Open Project of Guangxi Colleges and Universities Key Laboratory of Complex System Optimization and Big Data Processing (No. 2016CSOBDP0103) and the National Natural Science Foundation of China (Nos. 61472451).

PoS(ISCC 2017)010

1. Introduction

The popularization and application of network communication technology have made the electronic data security a very critical problem. Cryptography, an important technical means to protect information security, is very important in the network era. Widely used in the network, electronic pictures have following characteristics: huge amount of information, compressibility, similarity among adjacent pixels. Therefore, ordinary cryptography is not suitable for image encrypting. In order to solve this difficult problem, it is necessary to find new encryption methods.

Chaos is a nonlinear pseudo random phenomenon which is easily generated in a certain system. Chaos has a very intrinsic natural connection with cryptography, whose nature is like what Shannon [1] proposed in his encryption idea, and is especially suitable for image encryption with large amounts of data. Therefore, the image encryption technology based on chaos has attracted the attention of scholars [2-4]. People put forward many kinds of algorithms for encrypting images with different chaotic systems. Some algorithms in encrypting image use 1D discrete chaotic maps. They have high speed, but the security is not high enough [4-6]. Some algorithms in encrypting image use hyper-chaotic systems--they have good security, but the time cost is huge [7-8]. In this article, we put forward a new algorithm for encrypting images by combining Sine map with the generalized Arnold transformation, which greatly reduces the time overhead and also has high security.

2. New Proposed Algorithm for Encrypting Images

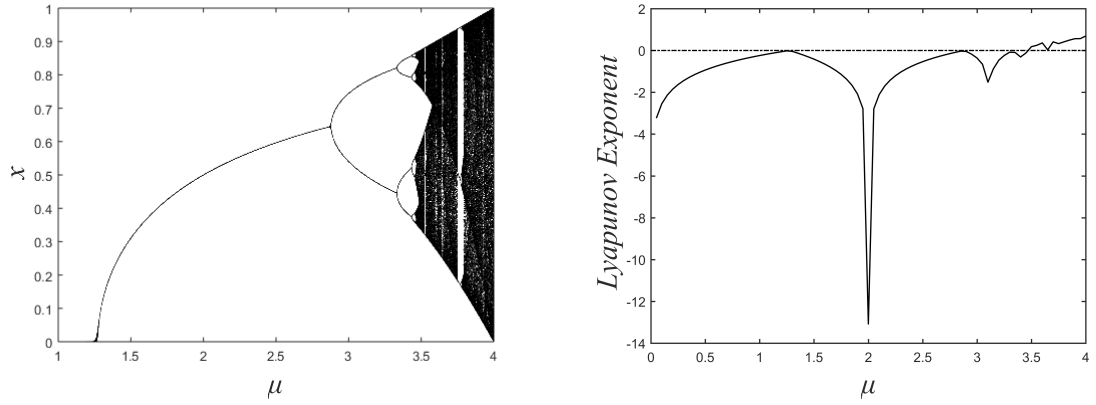
2.1 Chaotic Systems

2.1.1 Sine Map

The mathematical model of the Sine map is as follows:

$$x_{n+1}=\mu/4\times\sin\left(\pi\times x_n\right) \tag{2.1}$$

where μ is the parameter of the system, and x_n are state values of it. When $\mu \in (3.5, 4)$ and $x_0\in(0, 1)$, system (2.1) is chaotic and the output sequence by iterating system (2.1) has the characteristics of chaos. The bifurcation chart and the Lyapunov Exponent chart are drawn in Figure 1(a) and 1(b) respectively. From Figure 1, One can see that system (2.1) has a similar characteristic with the famous Logistic map, and it has a analogous chaotic feature with Logistic map. When μ> 3.5, the Lyapunov Exponent is positive, hence the Sine map system is chaotic within the range of parameter $\mu \in (3.5, 4]$.



(a) (b)

Figure 1: The bifurcation diagram and Lyapunov Exponent diagram of Sine map.

2.1.2 The Chaotic 2D Transformation

In this treatise, The Arnold chaotic 2D transformation was extended and a generalized Arnold cat map was introduced. We employ the generalized chaotic 2D transformation to permutate the image pixel coordinate location. Mathematically, the generalized Arnold map is describeed as follows

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} e & f \\ g & h \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \ mod \ M \tag{2.2}$$

When parameters set $\{e, f, g, h\}$ are several secret values in the 2D transformational matrix, (x_n, y_n), as well as (x_{n+1}, y_{n+1}), is the position coordinate or location of before transformation and after one time iterating transformation respectively, while M indicates that the picture has M rows as well as M columns of pixels in an image. When its determinant satisfies $(eh\text{-}fg) = 1$, Eq.(2.2) maps $(x_n, y_n) \in [0, M\text{-}1]$ to $(x_{n+1}, y_{n+1}) \in [0, M\text{-}1]$. It can be iterated t times, then we can transform the coordinate (x_0, y_0) to the coordinate (x_t, y_t). Here, (x_0, y_0) and (x_t, y_t) denote pixel positions in plain image and permutated image respectively.

2.2 Encryption and Decryption Process

2.2.1 Encryption Process

Consider an 8-bit gray picture with size $M \times M$ which will be encrypted. Let $N = M \times M$.

Step 1: Set the secretive key parameters of $\{x_0, \mu, e, f, g, h, t, C_0, \delta\}$, where μ parameter controls the Sine system (2.1) and x_0 determines its initial state value to Sine system (2.1), and $\{e, f, g, h\}$ determines the structure of system (2.2), t is a number of iterations for system (2.2) in permutation process.

Step 2: Input the plain text gray image which is an $M \times M$ matrix $\mathbf{P} = [\ P(i, j)\ ]$. Where, i and j are integers whose values belong to $[1, M]$.

Step 3: Do permutation operaton, and image $\mathbf{P}$ is permutated by iterating the chaotic map (2.2) for t times, and one can get a permutated image which can be expressed as $\mathbf{R}_{M \times M} = [R(i, j)]$.

Step 4: Convert the permutated image matrice $\mathbf{R}$ to the 1D vector $\mathbf{S} = [S(1), S(2), \ldots, S(N)]$.

Step 5: Iterate Eq. (2.1) for $(N+1000)$ times by using μ and x_0, and record the N values in the rear. We obtain the chaotic 1D array $\mathbf{X}$, which has N elements, $\mathbf{X} = [\ X(i)\ ]$, $i = 1, 2, \ldots, N$.

Step 6: Calculate the chaotic key-stream $\mathbf{K} = [\ K(i)\ ]$ by utilizing Eq.(2.3):

$$K(i) = \lfloor (X(i) \times 10^9) \rfloor \ mod \ 256, i = 1, 2, \cdots, N. \tag{2.3}$$

Step 7: Encrypt $\mathbf{S}$ by using Eq. (2.4) to obtain the cipher-image 1D sequence $\mathbf{C} = [\ C(i)\ ]$:

$$\begin{cases} C(i) = [(S(i) + C_0) \ mod \ 256] \oplus K(1), if \ i = 1, \\ C(i) = [(S(i) + C(i-1)) \ mod \ 256] \oplus K(i), i \in [2, N]. \end{cases} \tag{2.4}$$

Step 8: Shift pixels in $\mathbf{C}$ and obtain the processed array $\mathbf{Q}$ by the following Eq.(2.5).

$$\begin{cases} i' = i + \delta, if \ \ i + \delta \in [1, N], \\ i' = i + \delta - N, if \ \ i + \delta > N, \\ Q(i') = C(i), i \in [1, N], i' \in [1, N] \end{cases} \tag{2.5}$$

Step 9: Reshape the 1D vector $\mathbf{Q}$ to the 2D matrice, we get the final encrypted image.

2.2.2 Decryption Steps

In the procedure of decrypting a ciphered image, the manipulation steps are contrary steps of encrypting its corresponding plain-image. Firstly, for all $i \in [1, N]$ and $i' \in [1, N]$, do reverse shift operation for every pixel using the following Eq. (2.6):

$$\begin{cases} i=i^{'}-\delta, if\ \ i^{'}-\delta>0, \\ i=i^{'}-\delta+N, if\ \ i^{'}-\delta\leqslant 0, \\ C(i)=Q(i^{'}), i\in[1,N], i^{'}\in[1,N] \end{cases} \tag{2.6}$$

Secondly, do inverse diffusion operation using the following Eq. (2.7).

$$\begin{cases} S(i)=[C(i)\oplus K(i)-C_0]\ mod\ 256, if\ i=1, \\ S(i)=[C(i)\oplus K(i)-C(i-1)]\ mod\ 256, i\in[2,N]. \end{cases} \tag{2.7}$$

Thirdly, do inverse permutation operation using the generalized Arnold map.

3. Experimental Results

In our experimental tests, the secret keys are set as (μ=3.998, x_0=0.201, e=2, f=5, g=3, h=8, t=5, C_0=178, δ=516). We apply Matlab R2016b to run the above algorithm on a PC which has a 3.3 GHz CPU, 4 GB memory and the 64 bits Microsoft Windows 7 operating system. The plaintext image used in the experiments is the grey image lena.pgm, which comes from the standard test pictures database CVG.

3.1 Analysing Number of Secret Keys

The number of secret keys of a algorithm represents the number of different keys existing in the cryptosystem. If an algorithm has a key space size large enough, then it can resist exhaustive attack. Usually, key space should be more than 2^{100}. In the current scheme, the whole secret key set includes $\{\mu, x_0, e, f, g, h, t, C_0, \delta\}$. If the accuracy of the parameters $\{e, f, g, h\}$ is 10^2, μ and x_0 of Sine map are assumed as 10^{15}, and $1\leq t\leq 30$, $0\leq C_0\leq 255$, $1\leq C_0\leq 65535$, then the key space is $(10^{2\times4+15\times2})\times30\times256\times65535\approx2^{155}$ and the number is greater than 2^{100}. Hence, our proposed algorithm has plenty of secret keys, and it has the resistibility to exhaustive attacks.

3.2 Analysing Sensitivity

3.2.1 Sensitivity to Keys

A fine cryptosystem should possesses sensitivity to key parameters, namely, when the keys used in decryption are slightly different from the keys used in encryption, the plaintext image can not be decrypted correctly. In order to measure this features quantitatively, mean squared error (*MSE*) is used, which is defined as follows:

$$MSE=\frac{1}{M_1\times M_2}\times\sum_{i=1}^{i=M_1}\sum_{j=1}^{j=M_2}[I_p(i,j)-I_d(i,j)]^2. \tag{3.1}$$

Where $I_p(i,j)$ and $I_d(i,j)$ are two pixel values in a plain picture and its decrypted one in the same position, M_1 is the row number and M_2 is the column number. The *MSE* should be zero when a encrypted image is decrypted exactly. Some experimental results of *MSE* for several classical images are given in Table 1. From Table 1, one can perceive that our algorithm possesses sensitivity to secret keys.

Images	x_0 has a error of 10^{-10}	μ has a error of 10^{-10}	δ has a error of 1
lena.pgm	9065.96	9021.13	9007.51
einstein.pgm	7190.95	7242.84	7221.54
cameraman.pgm	9447.93	9493.07	9434.62
peppers.pgm	8233.16	8141.01	8132.32

Table 1: Some *MSE* Values of Several Tests

3.2.2 Sensitivity to Plaintext

To test sensitivity to plaintext, we introduce two index values $NPCR$ and $UACI$, which can be calculated by Eq. (3.2) and (3.3) respectively.

$$NPCR = \frac{1}{M_1 \times M_2} \times \sum_{i=1}^{i=M_1} \sum_{j=1}^{j=M_2} D(i,j) \times 100\%. \tag{3.2}$$

$$UACI = \frac{1}{M_1 \times M_2} \times \frac{\sum_{i=1}^{i=M_1} \sum_{j=1}^{j=M_2} [c_1(i,j) - c_2(i,j)]}{255} \times 100\%. \tag{3.3}$$

Where, $c_1(i,j)$ and $c_2(i,j)$ denote two pixel values in a couple of ciphered images which have one different pixel, $D(i,j)$ measures their dissimilarity between $c_1(i,j)$ and $c_2(i,j)$. If $c_1(i,j)$ is equal to $c_2(i,j)$ then $D(i,j)$ is equal to zero. If $c_1(i,j)$ is not equal to $c_2(i,j)$ then $D(i,j)$ is equal to one. We have calculated their *NPCR* and *UACI* about image Lena. By chosing 5 pixels at different positions in each plain Lena image, every time we modify the pixel value by adding 1 or subtracting 1, then calculate *NPCR* and *UACI* for the five cases according to Eq. (3.2) and Eq. (3.3). The calculation results are listed in Table 2, which show that our proposed algorithm has sensitivity to plain images.

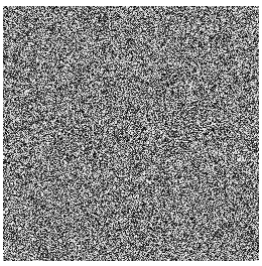
Positions	(1,1)	(64,1)	(128,128)	(212,18)	(256,256)
NPCR	99.56%	99.59%	99.59%	99.65%	99.62%
UACI	33.48%	33.48%	33.54%	33.62%	33.52%

Table 2: Tests of Sensitivity to Plain-images.

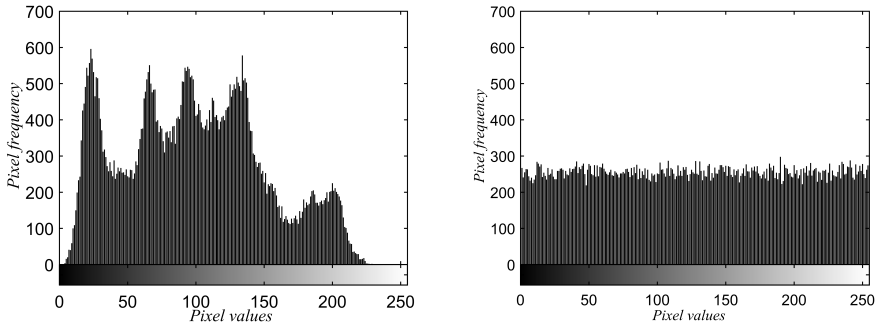
3.3 Distribution of the Ciphertext

The image gradation histogram depicts its grayscale distribution, and it exhibits the statistical information about an image. Figure 2(a) together with 2(b) depicts the plain image Lena along with its encryption one respectively, while Figure 2(c) and 2(d) depict the corresponding gradation histogram about plain image Lena along with its encryption one. It is descried from Figure 4 that the gradation distribution about the pixel values in plain image Lena is uneven. However, the gradation distribution concerning the pixel values corresponding to the ciphered image Lena is almost uniform. For example, the ciphered image can hide the statistical characteristics regarding to its plain image. Hence, it is able to resist statistical analysis attacks.

(a) The plain image Lena.



(b) The encrypted image Lena.



(c) The gradation histogram of plain Lena. (d) The gradation histogram of encrypted Lena.

Figure 2: Encryption Effect of the Proposed Algorithm.

3.4 Correlation Analysis for Each Adjacent-pixel Pair

A meaningful image usually has a high similarity to any two adjacent pixels, which is called as correlation. To enhance the resistivity to statistical attacks, a fine encrypted image should reduce the correlation as much as possible. As a experimental test, we select each group of adjacent-pixel-pair (about vertical, horizontal as well as diagonal directions) from the ciphered image Lena, and figure out their coefficients of correlation as Wang did in Refer.[6]. The results are listed in Table 3. Comparing to the Refs.[6], [8] and [9], our proposed algorithm has better results.

Directions	This paper	Ref.[6]	Ref.[8]	Ref.[9]
1-Horizontal	-0.000801	0.003011	0.0019	-0.0226
2-Vertical	-0.001984	-0.000762	0.0053	0.0041
3-Diagonal	0.002590	-0.005712	0.0042	0.0368

Table 3: The Correlation Coefficients for Lena Cipher-image

3.5 The Shannon's Information Entropy Analysis

Shannon's information entropy [10] is usually used to measure the randomicity of a message source. For an 8-bit image information source, the Shannon's entropy is expressed as

$$H(m)=-\sum_{i=1}^{i=256} P(m_i)\log_2[P(m_i)]. \tag{3.4}$$

where m_i is the ith gray-level value, $P(m_i)$ is the probability of value m_i emerging in the image. Evidently, for an 8-bit true random image, $P(m_i)$=1/256 and the entropy is 8. A well-encrypted image should have the information entropy very close to 8. The Shannon's information entropy values for encrypted image Lena by different algorithms are shown in the following Table 4. Compared with the results of Refs. [5,6,8,9], our algorithm has the maximum Shannon entropy.

This paper	Ref.[5]	Ref.[6]	Ref.[8]	Ref.[9]
7.997649	7.997158	/	7.9974	7.9973

Table 4: The Shannon Entropy for Cipher Image Lena

3.6 Complexity Analysis

The complexity of an image encryption algorithm is generally concerned with its time complexity. With regard to the time cost of the algorithm, each pixel is confused and diffused in turn. If the processing of each pixel is considered as a basic operation, the number of pixels is $N=M\times M$, then the proposed algorithm has the time complexity O(N). Through the above computing platform, the average time of encrypting or decrypting an 8-bit grey image which has a size of 256×256 is 0.188 second. The results state clearly that the speed of our proposed encryption and decryption algorithm is fast and acceptable.

4. Conclusion

This research elaborated an original algorithm for encrypting image data by combining Sine map with generalized Arnold transformation. The algorithm has the structure of permutation-diffusion-shifting. By doing the shifting process, one can avoid multiple rounds of permutation-diffusion iterations and achieve higher encryption strength, which can reduce the encryption time overhead. The results of security analysis and experimental tests for the proposed scheme have been given in detail. The analytical results as well as laboratorial test outcomes certified that our original algorithm possesses high degree of security, and it has a strong practical application potential in Internet-based image secure communication.

References

[1] C. E. Shannon.*Communication theory of secrecy systems*[J]. Bell System Technical Journal, 28(4):656-715(1949)

[2] X.J. Tong, M.G. Cui. *Image encryption with compound chaotic sequence cipher shifting dynamically*[J]. Image and Vision Computing, 26(6):843-850(2008)

[3] D. Xiao, X.F. Liao, P.C. Wei. *Analysis and improvement of a chaos-based image encryption algorithm*[J]. Chaos, Solitons & Fractals, 40(5):2191-2199(2009)

[4] G. J. Zhang, Q. Liu. *A novel image encryption method based on total shuffling scheme*[J]. Optics Communications, 284(12):2775-2780(2011)

[5] G. D.Ye, X.L. Huang. *An efficient symmetric image encryption algorithm based on an intertwining logistic map*[J]. Neurocomputing, 251: 45-53(2017)

[6] C.Q.Wang, X. Zhang, Z. M. Zheng. *An efficient image encryption algorithm based on a novel chaotic map*[J], Multimedia Tools and Applications, 76(22):24251-24280(2017)

[7] C. X. Zhu. *A novel image encryption scheme based on improved hyperchaotic sequences*[J]. Optics Communications, 285(1):29-37(2012)

[8] H. M.Yuan, Y. Liu, L.H. Gong, J. Wang. *A new image cryptosystem based on 2D hyperchaotic system*[J]. Multimedia Tools and Applications, 76(6):8087-8108(2017)

[9] L. Xu, X. Gou, Xu, Z. Li, J. Li. *A novel* chaotic *image encryption algorithm using block scrambling and dynamic index based diffusion*[J]. Optics and Lasers in Engineering, 91: 41-52 (2017)

[10] P. Zhen, G. Zhao, L. Min, X. Jin. Chaos-*based image encryption scheme combining DNA coding and entropy*[J]. Multimedia Tools and Applications, 75(11):6303-6319(2016)