Journal of Information Hiding and Multimedia Signal Processing ©2019 ISSN 2073-4212
Ubiquitous International Volume **10**, Number **2**, March **2019**

Image Encryption Algorithm based on Chaos and Its Implementation on FPGA

Junhua Zhu and Baoxiang Du*

Department of Electronic Engineering
Heilongjiang University
No. 74, Xuefu Road, Nangang District, Harbin, Heilongjiang, China
*Corresponding author: dubaoxiang@sina.com

Received July 2018; revised December 2018

Abstract. This paper proposes an improved practical chaotic image encryption algorithm based on the existing chaotic image encryption algorithm, this method not only Changes pixel value but also scrambles pixel position, so that its security is improved and its attack ability is strongfinally, the improved algorithm is implemented on FPGA(FieldProgrammable Gate Array). The experimental results show that the method has good encryption effect and has a certain practical application value in the domain of secure communication.
Keywords: Chaotic encryption; Secure communication; FPGA

1. **Introduction.** With the continuous development of communication technology, more and more data are transmitted through the network, and secure communication becomes more and more important. In the information encryption, the traditional encryption algorithms, such as DES(Data Encryption Standard) and AES(Advanced Encryption Standard), are designed for one dimensional data flow. It does not take into account that digital images and digital video have the characteristics of large data and strong correlation, and the efficiency of encryption is not high, and after encryption, it is likely to retain the general outline of the image, so it is not suitable for encrypting video and image information. The chaotic system has a sensitive dependence on the initial value, the so-called "Butterfly Effect", which makes it able to produce uncertain, non repetitive, and similar random results for different initial values, which is very suitable for encryption, scholars have put forward many image encryption algorithms based on chaotic system [1-12].

Literature [1] proposes an image encryption algorithm that uses Logistic chaotic system to change pixel values, literature [2,3] both defines a chaotic system and uses them to change pixel values, literature [4,5] uses mixed chaotic systems and spatiotemporal chaos to change pixel values respectively. literature [6] encrypts images using pseudo random sequences generated by piecewise nonlinear chaotic maps. literature [7] combines tent mapping with Logistic chaotic mapping, and proposes a new image encryption chaotic mapping to encrypt images. The disadvantage of these algorithms, which only change the pixel value, is that once attackers breaks through the encryption matrix, they can decode the encrypted image, in the same way, the encryption algorithm that only makes image pixel position disorder is easy to be broken by statistical attack or by comparing pixel values. In order to improve the security, the document [8-12] uses two different chaotic systems to encrypt the pixel position values and pixel values respectively. By referring to

the algorithm proposed in the above literature, This paper refers to the algorithms proposed in the literature and the encryption effects of different image encryption algorithms listed in the literature [13]. and with the help of FPGA hardware circuit with parallel computing capability, the improved method is realized.

2. **Chaotic encryption algorithm.** The algorithm in this paper consists of two parts, the changes of pixel values by the improved logistic mapping and the scrambling of pixel positions by Arnold mapping. The specific encryption process is shown in Figure 1. Decryption is the inverse operation of encryption, and the decryption process is shown in Figure 2.

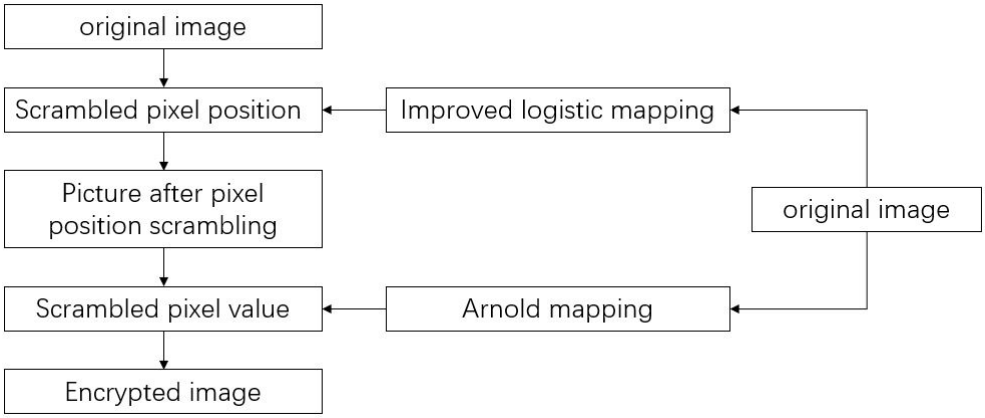

FIGURE 1. Encryption process

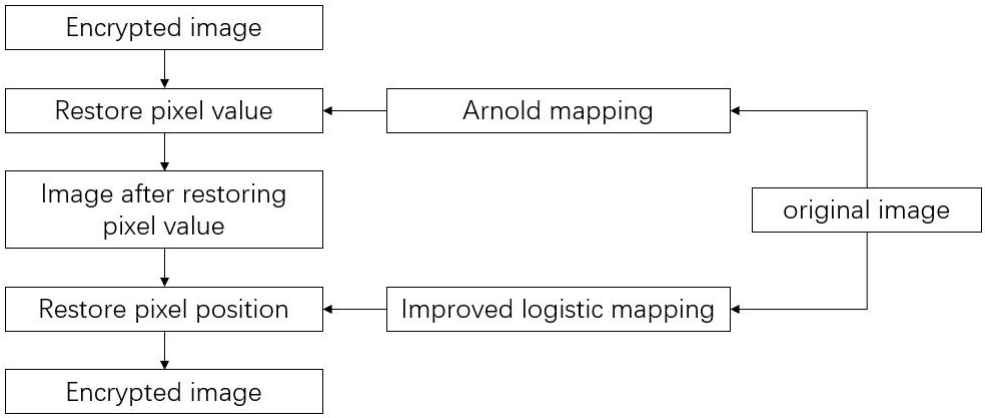

FIGURE 2. Decryption process

2.1. **Improved logistic mapping.** The logistic mapping which is improved is unimodal. Its expression is :

$$x_{k+1} = ux_k(1 - x_k) \tag{1}$$

where: x_k is the starting value of the iteration, and $x_k \in [0, 1]$, u is the system parameter, and $u \in [0, 4]$.

Literature [14] has done a lot of analysis of the encryption performance of formula (1), proving that it has good encryption performance, but, according to the ergodicity of chaos, with the increasing number of iterations, the chaotic sequence will evolve into a periodic

sequence, which will affect the encryption effect. In order to improve the encryption effect of the chaotic sequence, the logistic mapping is improved by mutual coupling. Two different logistic mappings are coupled with each other. The first chaotic map provides initial values for second chaotic maps, and second chaotic maps provide initial values for the first chaotic map, which can increase the number of iterations of chaotic sequences evolving into periodic sequences. The corresponding expression is :

$$\begin{cases} x_{k+1} = u_1 x_k(1 - x_k) \\ x_{k+1} = u_2 x_k(1 - x_k) \end{cases} \tag{2}$$

Where: u_1, u_2 take different values, and $u_1, u_2 \in [0, 4]$, the specific method is shown in figure 3, x is the input value and y is the output value.

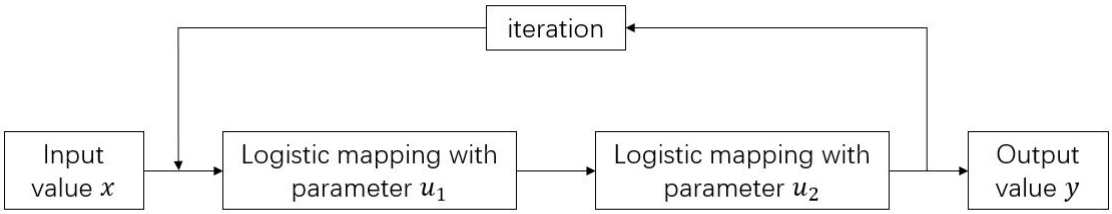

FIGURE 3. Improved chaotic sequence generation method

Taking into account the size of the key space, 3 sets of Logistic chaotic maps with different parameters are alternatively iterated to generate chaotic sequences. The specific method is to use the logistic chaotic map of that set of parameters according to the size of the substituted values. When the input value is $0 \leq x_m \leq 0.33$, the first set of chaotic maps with parameters u_1 and u_2 are used for iteration, when the input value is $0.33 < x_m < 0.66$, the second set of chaotic maps with parameters u_3 and u_4 are used for iteration, when the input value is $0.66 \leq x_m \leq 1$, the second set of chaotic maps with parameters u_5 and u_6 are used for iteration, the input value of the next iteration is the output value of the last operation. Assuming that the input value is x_m and the output value is y_n, the specific steps are shown in figure 4.

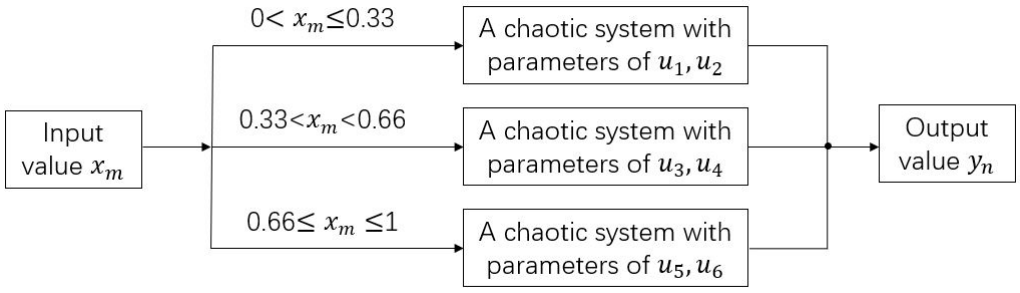

FIGURE 4. Alternate iterative process

2.2. **Arnold transformation.** Arnold transform is a chaotic mapping from the torus surface to itself, also known as Cat mapping, it was proposed by the Russian mathematician V.J.Arnold, Because it often uses a picture of a cat's face to demonstrate it, so it gets a name. It has many excellent mathematical properties [15]: (1) reversible; (2)

Unchangeable area ; (3) Ergodicity and miscibility ; (4) Topological mobility ; (5) The set of periodic orbital points on the torus is dense ; (6) The only hyperbolic fixed point.

The specific expression for the Arnold transform is :

$$\begin{cases} x_{n+1} = (x_n + y_n) mod 1 \\ y_{n+1} = (x_n + 2y_n) mod 1 \end{cases} \tag{3}$$

Where : $mod1$ means only a decimal part is taken, That is, $xmod1 = x - [x]$, therefore, the phase space of $[x_n, y_n]$ is limited to the unit square. The formula (3) can be expressed in matrix form as follows:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} mod1 = C \begin{pmatrix} x_n \\ y_n \end{pmatrix} mod1 \tag{4}$$

where: $\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$ is a transformation matrix, its determinant value is 1, so this mapping is a one to one mapping with a constant area and no attractors, any point in a unit square is uniquely transformed into another point in the unit square. Cat mapping has two very typical characteristics of chaotic phenomena: Stretch (multiply the matrix C to make the value of x, y larger) and fold (take the mode mod1 to make x and y back in the unit rectangle). By calculating the eigenvalues of the matrix C, we obtain two Lyapunov indices of the cat mapping, namely $\ln(0.5(3+\sqrt{5}))$ and $\ln(0.5(3-\sqrt{5}))$. It turns out that cat mapping is a chaotic mapping.

Geometrically, its phase space can be extended from $[0;1] \times [0;1]$ to $0.1; ...; N-1 \times 0; 1; ...; N-1$, so the discretized cat mapping expression is obtained.

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} modN \tag{5}$$

Because the state space of the discrete cat map is limited, it may no longer have good chaotic characteristics. But it still has the stretching and folding property of cat mapping geometrically, this leads to two adjacent points $(i; j)$ and $(i; j+1)$ no longer adjacent after multiple discrete transformation iterations, indicating that the transformation has certain initial value sensitivity. By using this property, the position of adjacent pixels of an image can be Scrambled, making it impossible to obtain relevant information of the original image from the image, so as to achieve the purpose of keeping confidential image information.

2.3. **The principle of encryption and decryption.** Encryption and decryption are implemented using improved logistic mapping and Arnold mapping. Logistic mapping is used to change and encrypt pixel values, and Arnold mapping is used to scramble and encrypt pixel positions. Since the algorithm is symmetric, decryption is the inverse of encryption.

2.3.1. *Change the pixel value.* The improved logistic map is used to encrypt the pixel value of the imagethe specific steps are as follows:

Step1. Set the control parameters of chaotic system $u = \{u_1, u_2, u_3, u_4, u_5, u_6\}$ and the initial value of X, logistic mapping was first iterated K times to eliminate the initial state effect, Continue to iterate 256 times to generate a chaotic sequence S of length 256, where $S = \{s_1, s_2, s_3, s_4, s_5, s_6, ...s_{m \times n}\}$.

Step2. Let $i = 1, 2, 3, 256$, s_i is the ith real value in the chaotic sequence, take the real number s_i from 3 to 10 decimal places, Using these 8 digits to make up a sequence of W, where $W = \{w_1, w_2, w_3, ..., w_8\}$, if the chaotic sequence value is $s_i = 0.1079652013$, then the sequence $W = \{7, 9, 6, 5, 2, 0, 1, 3\}$ is generated.

Step3. Turning the image into a grayscale image, if the image is a grayscale image, there is no need for this step. The pixel value p_i in image P is divided into one group for every 256 pixels, if the last group is less than 256, make up for it with 0. convert them into binary, generating array $p_{Bit} = \{Bit_1, Bit_2, Bit_3, ..., Bit_8\}$. For example, if the pixel value $p_i = 186$, then $P_{Bit} = \{1, 0, 1, 1, 1, 0, 1, 0\}$.

Step4. The sequence W obtained in step 2 is arranged from large to small, (if the two values are equal, they are arranged in order), get ordered sequence w_l, the array W_t is then used to hold the position of the W_l elements in w, If $W = \{7, 9, 6, 5, 2, 0, 1, 3\}$, the rearranged sequence is $W_l = \{9, 7, 6, 5, 3, 2, 1, 0\}$, and the position sequence is $W_t = \{2, 1, 3, 4, 8, 5, 7, 6\}$.

Step5. Using 256 chaotic sequence values of the location sequence $W_t = \{T_1, T_2, T_3, ..., T_8\}$ scramble the pixel values of each group $p_{Bit} = \{Bit_1, Bit_2, Bit_3, ..., Bit_8\}$, a new form of rearrangement after pixel value scrambling, that is $p_{BBit} = \{BBit_1, BBit_2, BBit_3, ..., BBit_8\}$, (Each set of pixel values from 1 to the 256 correspond to the chaotic sequence values). The specific steps are as follows, the numbers in $p_{Bit} = \{Bit_1, Bit_2, Bit_3, ..., Bit_8\}$ are swapped in the order of the corresponding $W_t = \{T_1, T_2, T_3, ..., T_8\}$, for example, if $P_{Bit} = \{1, 0, 1, 1, 1, 0, 1, 0\} W_t = 2, 1, 3, 4, 8, 5, 7, 6$, the binary arrangement of the point after bit scrambling will become $P_{Bit} = \{0, 1, 1, 1, 0, 1, 1, 0\}$.

Step6. Convert the resulting ith - point binary number to decimal, and get the ith - point encrypted message.

Step7. Repeat 2 6 steps $m \times n$ times to get the complete encrypted ciphertext.

2.3.2. *Scramble pixel position.* The Arnold map is used to Scramble the pixel position of an image that has already changed its pixel valuethe operation of pixel position scrambling stage will realize the global scrambling of image pixel position to break the correlation of adjacent pixels. This algorithm uses the Arnold mapping in formula (5) to Scramble the position of image pixels. Where the input and output of the formula are all integers, for $M \times N$ images, the specific scrambling method is:

Step1. Set up a large output space equal to the original matrix.

Step2. Substitute each pixel coordinate (i, j) in this space as the initial value (x_0, y_0) into equation (5) successively, new coordinates are generated by chaotic iteration as (x_1, y_1). (the number of iterations is variable n, and the number of iterations is determined according to the specific circumstances.).

Step3. Put the first (x_1, y_1) pixel in the input space into the (x_0, y_0) position in the output space, namely (i, j) position, when the whole space is filled, get the final ciphertext.

3. **Implement with FPGA.** As a kind of hardware that can realize parallel operation, FPGA can improve the processing speed of data and improve the processing efficiency when dealing with digital information such as video images with a large number of same structure data.

3.1. **FPGA system design.** In order to implement the encryption algorithm proposed in this paper on FPGA. The encryption system based on FPGA is designed with the EP4CE75F23C8 chip of ALTERA company. The system diagram is shown in Figure 5.

The encryption system based on FPGA includes SD(Secure Digital) card controller, SDRAM(Synchronous Dynamic Random Access Memory) controller, LCD(Liquid Crystal Display) controller, Arnold encryption module and logistic encryption module. The overall flow of the encryption system is as followsStoring image data in SD cardWhen the whole system begins to work, the image data is read out from the SD card first, and then the image data is temporarily cached in the SDRAM through the SDRAM interface control module, read the cache data from SDRAM, and conduct pixel value scrambling

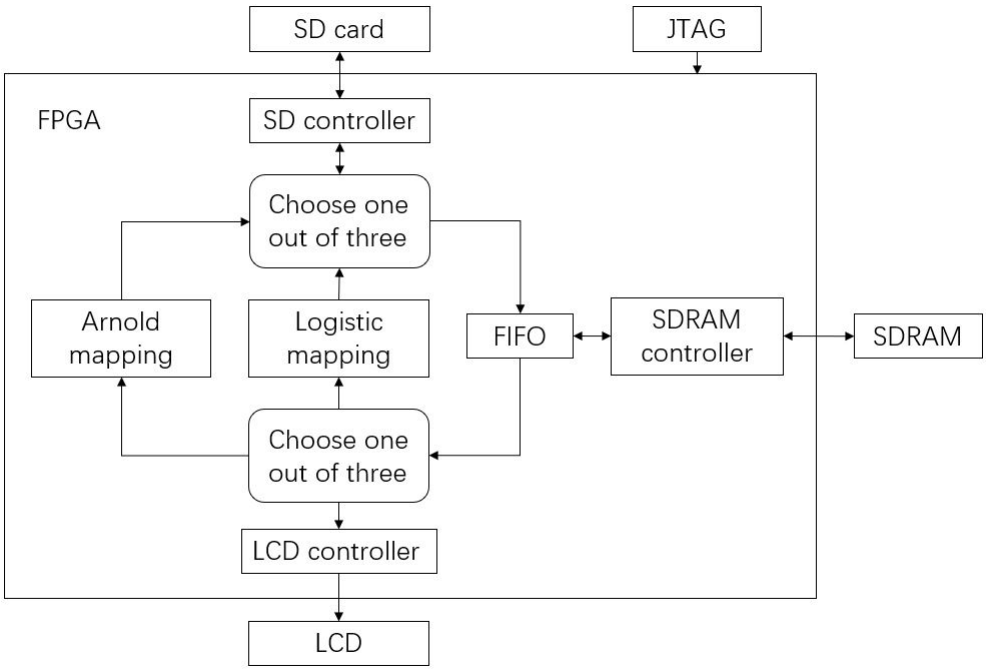

FIGURE 5. Encryption system structure diagram

encryption on image data through Logistic encryption module, the data encrypted by Logistic encryption module will be temporarily cached in SDRAM, read the cache data from SDRAM, and use the Arnold encryption module to Scramble the pixel position of the image data. After encryption is completed, it is sent to the LCD control module to complete the LCD display.

Verilog HDL is used to implement the software part of this system. SD card controller, SDRAM controller, LCD controller, Arnold encryption module and logistic encryption module are designed.

3.2. **Experimental results of encryption system.** The implementation result of the encryption system based on FPGA is shown in Figure 6.

4. **Cryptanalysis.** According to the modern cryptography principle, to ensure the security of encrypted information, the following points must be ensured: First, it has enough key space; Secondly, extremely sensitive to both plaintext and ciphertext; Thirdly, ciphertext should be uniformly distributed and adjacent data irrelevant.

4.1. **Key space analysis.** The size of the key space directly affects the security of the password system, a large key space can effectively resist exhaustive attack. In this algorithm, the initial values x_0 and parameters $u_1u_2u_3u_4u_5u_6$ in the logistic system were selected as keys. Double precision data is adopted for the key. When 32bit computer is used, the double precision key is 64bit. In ignore other parameter values, the key space is $2^{64} \times 2^{64} \times 2^{64} \times 2^{64} \times 2^{64} \times 2^{64} \times 2^{64} = 2^{448}$, Such a large key space is sufficient to resist exhaustive attack.

4.2. **Sensitivity analysis.** The higher the sensitivity of ciphertext to plaintext, the better the algorithm can resist differential attack. The basic idea of differential attack is to recover some key bits by analyzing the effect of the difference between plaintext pairs on the difference between ciphertext pairs,if the ciphertext image changes greatly due

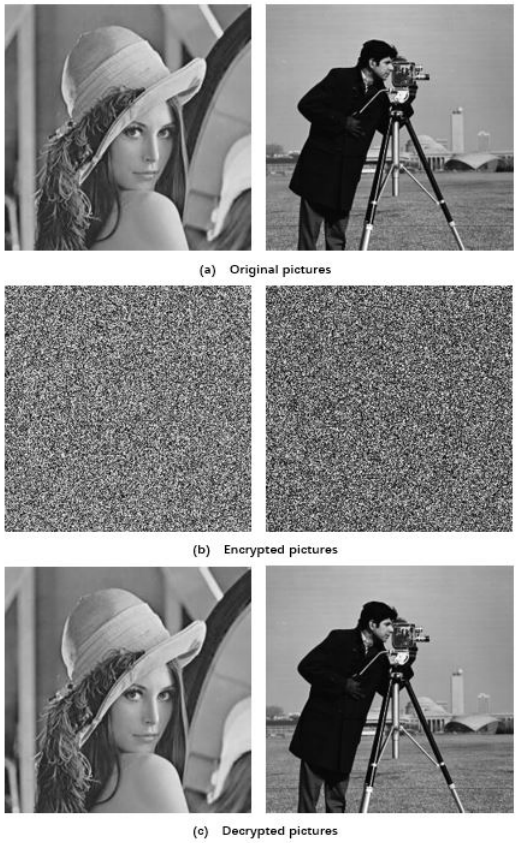

FIGURE 6. Original image, encrypted image and decrypted image

to a small change in the plaintext image, the difference attack is invalid. In this paper, we introduce NPCR(number of pixels change rate) and UACI(unified average changing intensity) to measure the sensitivity of encryption algorithm to plaintext. When there is only one pixel value difference between two plaintext images, let the pixel values of point (i,j) in their ciphertext images be $c_1(i,j)$ and $c_2(i,j)$ respectively. if $c_1(i,j) = c_2(i,j)$, $Q(i,j) = 0$, if $c_1(i,j)c_2(i,j)$, $Q(i,j) = 1$. NPCR was used to measure the change rate of pixels between two images. The specific formula is :

$$NPCR = \frac{\sum_i \sum_j Q(i,j)}{M \times N} \times 100\% \tag{6}$$

UACI is used to measure the average change intensity of pixels. The specific formula is :

$$UACI = \frac{1}{M \times N}[\sum_i \sum_j \frac{\mid c_1(i,j) - c_2(i,j) \mid}{255}] \times 100\% \tag{7}$$

The ideal expected value formulas of NPCR and UACI are respectively:

$$NPCR_E = (1 - 2^{-n}) \times 100\% \tag{8}$$

$$UACI_E = \frac{1}{2^{2n}} \times \frac{\sum_{i=1}^{2^n-1} i(i+1)}{2^n - 1} \times 100\% \tag{9}$$

Where, M and N are the number of rows and columns of image pixels respectively, and n is the bits depth of image color. For 8-bit grayscale images ($n = 8$), the ideal expectation values of NPCR and UACI are: $NPCR_E = 99.6094\%$, $UACI_E = 33.4635\%$. In this

experiment, 8 groups of 8-bit grayscale images (n=8) were selected for encryption, each group has two pictures. One of them is the original image, the other is to change the image of a random grayscale in the original image. The values of NPCR and UACI between 8 groups of ciphertext images are obtained respectively, and the results are shown in Table 1. The results show that the values of NPCR and UACI obtained from experiments are close to ideal values. This shows that the encryption scheme is very sensitive to the slight change of the plaintext image, that is, the algorithm proposed in this paper has good ability to resist differential attack.

TABLE 1. Comparison of NPCR and UACI values with ideal values obtained from 8 sets of experiments

	Ideal values	First group	Second group	Third group	Fourth group	Fifth group	Sixth group	Seventh group	Eighth group
NP CR	99.609 37%	99.603 56%	99.569 74%	99.597 81%	99.604 72%	99.587 21%	99.567 12%	99.641 36%	99.643 17%
UA CI	33.463 54%	33.427 15%	33.655 17%	33.692 14%	33.590 24%	33.616 37%	33.673 54%	33.596 45%	33.514 21%

4.3. **Analysis of statistical properties of algorithms.**

4.3.1. *Correlation of adjacent pixels.* The adjacent pixels of the image are highly correlated, so the encrypted image should destroy the correlation of the plaintext image as much as possible to resist the statistical attack. In order to test the correlation between the adjacent pixels of the plaintext image and the ciphertext image, select the adjacent pixels (horizontal, vertical and diagonal) of N in the image, and then use the following formula to quantify the correlation coefficient of the adjacent pixels [16].

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

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{11}$$

$$Conv(x,y) = \frac{1}{N}\sum_{x_i - \bar{y}} \tag{12}$$

$$\gamma_{xy} = \frac{Conv(x,y)}{\sqrt{Dx}\sqrt{D(y)}} \tag{13}$$

Where$X$ and y respectively represent the grayscale values of the two adjacent pixels in the imageγ_{xy} is the correlation coefficient of 2 adjacent pixels. In the experiment, 1800 adjacent pairs of pixels in horizontal, vertical and diagonal pairs are tested respectively, and the results are shown in Table 2. It can be seen that the adjacent pixels before encryption have a strong correlation with the correlation coefficient close to 1, after encryption, the correlation coefficient is small, encryption damages the correlation.Table 2 also lists the correlation coefficients of adjacent pixels before and after encryption by other algorithms. Through comparison, we can find that the algorithm in this paper has a good encryption effect.

1800 sets of adjacent pixel pairs are randomly selected from the image to be measured, the vertical correlation distribution of these pixel pairs before and after encryption is

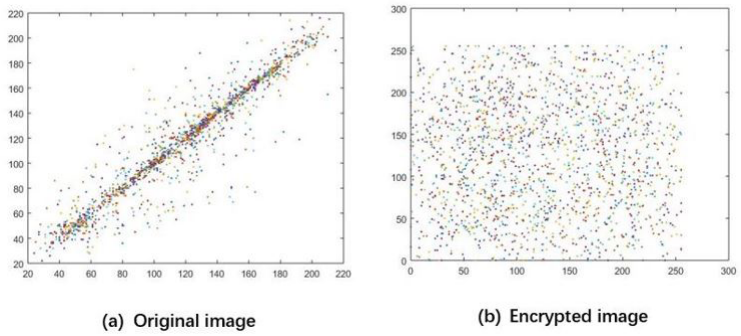

FIGURE 7. Adjacent pixel correlation of original image and encrypted image

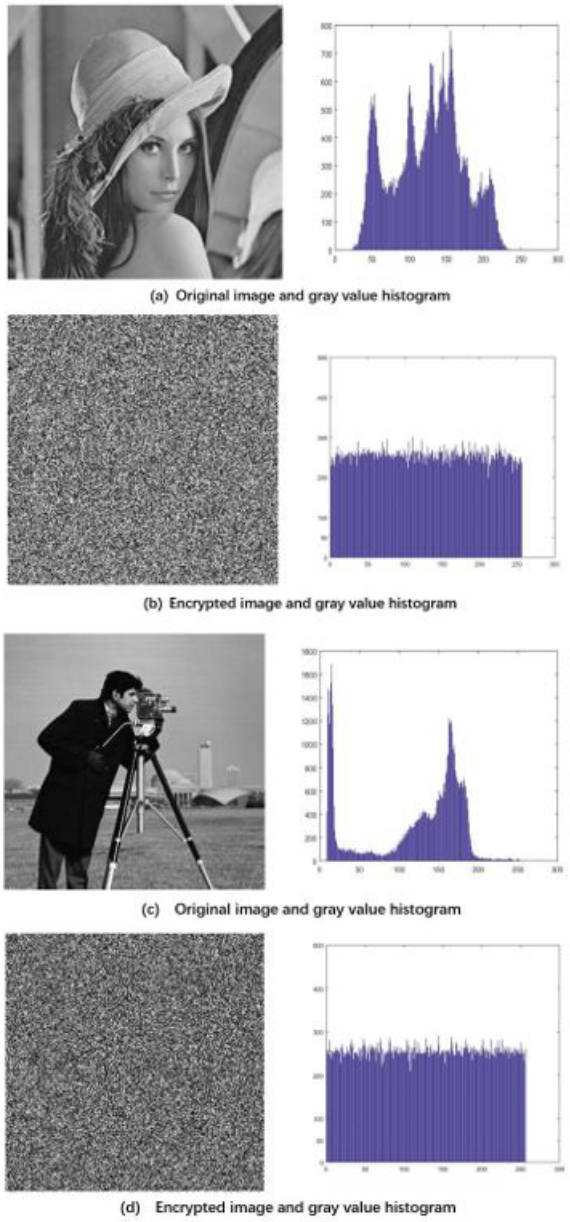

FIGURE 8. images and gray value histogram

TABLE 2. Correlation coefficient of adjacent pixel pairs before and after image encryption

		Horizontal	Vertical	Diagonal
This paper	Original image	0.97281	0.96583	097835
	Cipher image	0.00681	0.00562	0.00376
literature [2]	Original image	0.96592	0.94658	0.92305
	Cipher image	0.0055002	0.0041189	0.0002136
literature [6]	Original image	0.986190	0.975640	0.965895
	Cipher image	-0.0228872	0.014593	0.036581
literature [7]	Original image	0.935241655	0.945623588	0.9254968747
	Cipher image	0.009642154	0.03421542	0.0205742611
literature [9]	Original image	0.987388	0.988765	0.983516
	Cipher image	0.005126	0.003467	0.001251
literature [10]	Original image	0.9450	0.9704	0.9247
	Cipher image	0.0017	-0.0003	0.0001

shown in figure 7, Where the horizontal and vertical coordinates are respectively pixel values of two adjacent points, figure 7 (a) represents the correlation of the original graph. Figure 7 (b) represents the correlation of the encrypted image. It can be seen from the figure that most of the plaintext image pixels appear on the diagonal, indicating a strong correlation between adjacent pixels in the image In the ciphertext image, the pixel distribution is more uniform, so we can see that the algorithm can effectively destroy the correlation of adjacent pixels.

4.3.2. *Gray value change.* Figure 8 (a), (c) and (b) and (d) respectively provide gray histogram of the original image and the encrypted image. Therefore, the encryption process has changed the uneven distribution of the original image pixel value into a relatively uniform distribution. that is, the statistical characteristics of plaintext images are broken, which reduces the correlation between plaintext and ciphertext and hides the statistical characteristics of images.

5. **Conclusion.** In this paper, an image encryption algorithm based on chaotic system pixel value change and pixel position scrambling is proposed and analyzed, the algorithm is analyzed and implemented on hardware. The experimental results show that the algorithm has the characteristics of large key space, good encryption effect and easy implementation, it has certain use value in the field of secure communication.

Acknowledgment. This work is partially supported by Science and technology research projectof Heilongjiang Provincial Education Department (NO.12541637) and graduate innovation research fund of Heilongjiang University (YJSCX2018-147HLJU)The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers,which have improved the presentation.

REFERENCES

[1] H.M. Xie, L.Xia and M.Y. Li, Image encryption system based on logistic chaotic mapping and FPGA implementation, *Aero Weaponry*, no. 2, pp. 56-60, 2016.

[2] C.Y. Song, Y.L. Qiao and X.Z. Zhang, An image encryption scheme based on new spatiotemporal chaos, *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 18, pp. 3329-3334, 2013.

[3] C. nal, K. Sezgin, P. Ihsan and Z. Ahmet, Secure image encryption algorithm design using a novel chaos based S-Box, *Chaos, Solitons and Fractals: the interdisciplinary journal of Nonlinear Science, and Nonequilibrium and Complex Phenomena*, vol. 95, pp. 92-101, 2017.
[4] S. Behnia, A. Akhshuni and H. Mahmodi, A novel algorithm for image encryption based on mixture of chaotic maps, *Chaos Solitons Fractals*, vol. 35, no. 2, pp. 408-419, 2008.
[5] S.G. Lian, Efficient image or video encryption based on spatiotemporal chaos system, *Chaos Solitons Fractals*, vol. 40, no. 5, pp. 2509-2519, 2009.
[6] N.Oussama, N.Lemnouar, Design of a tweakable image encryption algorithm using chaos-based schema, *International Journal of Information and Computer Security*, vol. 8, no. 3, pp. 205-220, 2016.
[7] S. Fajeuddran, M. Doraipandian, A nonliner two dimensional logistic-tent map forsecure image communication, *International Journal of Information and Computer Security*, vol. 10, no. 2/3, pp. 201-215, 2018.
[8] E. Yavuz, R. Yazici, M.C. Kasapba, E. Yama, A chaos-based image encryption algorithm with simple logical functions, *Computers and Electrical Engineering*, vol. 54, pp. 471-483, 2016.
[9] X.Y. Wang, S.X. Gu and Y.Q. Zhang, Novel image encryption algorithm based on cycle shift and chaotic system, *Optics and Lasers in Engineering*, vol. 68, pp. 126-134, 2015.
[10] W.C. Ci, Q. Wang, F.M. Huang, Z.S. Yuan and C.S. Tao, Image adaptive encryption algorithm based on affine and compound chaos , *Journal on communication*, vol. 33, no. 11, pp. 119-127, 2012.
[11] G. Xu,Y.D. Zhang, X.X. Zhang and X.J. Zhang, Image encryption algorithm based on alternate iteration chaotic system, *Chinese Journal of Engineering*, vol. 34, no. 4, pp. 464-470, 2012.
[12] X.H. Deng, C.L. Liao, C.X. Zhu and Z.G. Chen, Image chaotic encryption algorithm with dual scrambling of pixel position and bit, *Journal on communications*, vol. 35, no. 3, pp. 216-223, 2014.
[13] J.Gayathri, S.Subashini, A survey on security and efficiency issues in chaotic image encryption, *International Journal of Information and Computer Security*, vol. 8, no. 4, pp. 347-381, 2016.
[14] G.Q. Cai, G.W. Song and D.P. Yu, Progressive watermarking techniques using genetic algorithms, *logistic mapping performance analysis of chaotic spread spectrum sequences*, vol. 21, no. 1, pp. 60-63, 2000.
[15] C.M. Wu, Improvement of discrete Arnold transform and its application in image scrambling encryption, *Acta Physica Sinica*, vol. 63, no. 9, pp. 91-110, 2014.
[16] G.R. Chen, Y.B. Mao and K. Charles, A symmetric image encryption scheme based on 3D chaotic cat maps, *Chaos, Solitons & Fractals*, vol. 3, no. 21, pp. 749-761, 2004.