Hindawi
Security and Communication Networks
Volume 2021, Article ID 6675565, 15 pages
https://doi.org/10.1155/2021/6675565

Research Article

A Novel Hyperchaotic Image Encryption System Based on Particle Swarm Optimization Algorithm and Cellular Automata

Jie Zeng and Chunhua Wang

College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

Correspondence should be addressed to Chunhua Wang; wch1227164@hnu.edu.cn

Received 17 December 2020; Revised 12 January 2021; Accepted 21 January 2021; Published 5 February 2021

Academic Editor: Guodong Ye

In this paper, we propose a hyperchaotic image encryption system based on particle swarm optimization algorithm (PSO) and cellular automata (CA). Firstly, to improve the ability to resist plaintext attacks, the initial conditions of the hyperchaotic system are generated by the hash function value which is closely related to the plaintext image to be encrypted. In addition, the fitness of PSO is the correlation coefficient between adjacent pixels of the image. Moreover, On the basis of hyperchaotic system, cellular automata technology is adopted, which can enhance the randomness of population distribution and increase the complexity and diversity of the population so that the security of the encryption system can be improved and avoid falling into local optimum. The simulation results and security analysis of the proposed encryption system demonstrate that the hyperchaotic image encryption system has high resistance against plaintext attack and statistical attack.

1. Introduction

In an era of rapid network science and the multimedia industry progress, using network for information transmission has become the primary choice for most people. As an important carrier of information, digital images are transmitted or stored through public channels. Therefore, the importance of information security is valued. It has been found that the traditional data encryption algorithm is not efficient and has many technical defects due to large data volume in digital images, high redundancy, and strong correlation between adjacent pixels. The existing research results show that the image encryption based on chaos theory have better characteristics than the traditional encryption algorithm [1, 2].

Many papers have been published on chaos generation to data [3–18]; hyperchaos is better for image encryption and security than chaos because of its high dynamic complexity. The concept of hyperchaos was first proposed by Rossler in 1979 to describe the characteristics of chaotic systems with two or more positive Lyapunov exponents [19]. This means that when compared with a chaotic system with only one positive Lyapunov exponent, the hyperchaotic system has the dynamics which can be extended in multiple directions simultaneously, resulting in more complex chaotic attractors and dynamical behaviors. The progression of hyperchaotic systems within unique cryptographic characteristics such as large key space and high sensitivity to initial values highlights its role in image encryption.

To improve the effectiveness of digital image encryption, researchers have proposed many new image encryption systems based on hyperchaotic systems [20–58]. For example, Khan proposed an image encryption scheme based on multiple chaotic S-boxes [20]. The scheme does not require multiple round keys and can be applied to high-speed communication systems. Liu et al. utilized a new four-dimensional hyperchaotic system to generate a key stream and displace and replace the plain image pixels in order to obtain a better pixel diffusion effect and higher encryption security [21]. However, those researchers found that the image encryption scheme based on S-box structure has the disadvantages of small key space and simple encryption structure. Image encryption schemes based on bit-plane and bit-level were proposed to compensate these shortcomings

[28–33]. Zhang et al. utilized chaotic systems to design a random visiting mechanism to the bit level of the plain image [29]. However, Wu discovered that the image encryption scheme based on three-dimensional bit matrix permutation was not suitable for secure communication. For this reason, an improved encryption scheme was proposed which ensured that the parameter values in chaotic systems depend not only on the key but also on the plain image [33].To solve the problems of single DNA coding rules and low sensitivity of chaotic encryption algorithm to key, the image encryption schemes based on DNA computing were introduced [34–41]. Hu et al. [36] proposed an image encryption scheme using a high-dimensional hyperchaotic system to generate the key stream and control the DNA coding rules, so as to achieve the effect of random coding. In [38], a new hyperchaotic image encryption system based on dynamic DNA encryption was proposed. In this encryption system, a pixel random diffusion mechanism based on plaintext is designed under DNA coding rules and binary operation rules. Wan et al. applied the diffusion algorithm to the plain image, then segmented the middle cipher image, and finally formed the final encrypted image by DNA operations [41]. Moreover, some researchers have invented hyperchaotic image encryption system based on cellular automata (CA) [42–46]. Chai et al. proposed an image encryption scheme based on memory hyperchaotic system, cellular automata, and DNA sequence computing [44]. Niyat et al. proposed a hyperchaotic image encryption scheme based on the heterogeneous cellular automata framework which consists of confusion and diffusion steps [45]. Nevertheless, Li et al. found that the scheme could not effectively resist the chosen-plaintext attack [46]. In [58], Ahmad et al. designed an image encryption system based on particle swarm optimization algorithm and low-dimensional mapping. However, the system has not enough large key space and has small data range. Therefore, strong pseudorandomness and ergodicity are not fully possessed by these generated data, thus resulting in low security. Besides, the problem of premature convergence and diversity of the population decreases in the PSO often lead to local optimum traps.

The purpose of this paper was to address the problems above by developing a new hyperchaotic image encryption system based on particle swarm optimization algorithm (PSO) and cellular automata (CA). To improve the ability to resist plaintext attacks, the initial conditions of the hyperchaotic system are generated by the hash function value which is closely related to the plaintext image to be encrypted. In addition, the fitness function of PSO is used to calculate the correlation between adjacent pixels of the image. By introducing CA technology, the complexity and diversity of the population in PSO are increased in order to avoid the loss of diversity of the population in the search space and enhance the security of the encryption system. The simulation experiments and security analysis reveal that the proposed system has a better encryption effect and superior security performance.

The paper is organized as follows. Section 2 introduces the relevant knowledge. In Section 3, a novel cryptosystem is proposed and its characteristics are analyzed. Simulation results and security analysis are provided in Section 4. Finally, Section 5 presents the conclusions drawn from this work.

2. Preliminary Knowledge

2.1. Hyperchaotic System. In this paper, a novel hyperchaotic image encryption system based on PSO and CA is proposed and a hyperchaotic system is adopted [59]:

$$\begin{cases} \dot{x} = a\left(y - bx^3 - (1+c)x\right), \\ \dot{y} = x - y + z, \\ \dot{z} = w - dy - ez, \\ \dot{w} = yz - fx, \end{cases} \tag{1}$$

where x, y, z, and w are state variables and a, b, c, d, e, and f are real constant system parameters of the chaotic system (1). When $a = 30$, $b = 0.03$, $c = -1.2$, $d = 50$, $e = 0.32$, and $f = 0.1111$, the system is hyperchaotic. To analyze its hyperchaotic behavior, the initial condition is set as (0, 0.1, 0.3, 0.4), and the time step of iteration is 0.001. Figure 1 shows the phase diagram of the system.

2.2. Particle Swarm Optimization Algorithm (PSO). PSO was proposed by Eberhart and Kennedy in 1995, inspired by the foraging of birds and fish [60]. PSO is a speculative calculation method based on swarm intelligence. This similarity can be explained by the assumption that a group of birds is foraging randomly in a large field, and no bird will know the position of the food in the field, but they only know the distance from the food. One of the most effective ways to find food is to hang out with the birds closest to the food site. PSO has the optimal understanding ability of social behavior, so it can be used to solve various optimization problems.

PSO has been widely recognized in signal processing, machine learning, adaptive control, fuzzy system control, neural network, function optimization, model classification, and other fields because of its simple implementation and intuitive process. Another advantage of PSO is that it involves only simple calculations and optimizes the problem by repeatedly trying to update the solution space associated with a particular fitness function.

In PSO, each bird represents a separate solution, which is called a "particle". The algorithm starts with the initial state of any particle and uses two optimal values for each particle's update: the first variable is the best value which the particle has achieved so far, called as pbest; the second is the gbest, refers to the best value in the all particles pbest values. All particles have three characteristics: fitness value, position, and velocity, which guide particles to find the best state of individuals and populations in the search space. The general PSO is described as follows:

The velocity and position of each particle are updated according to the following equations:

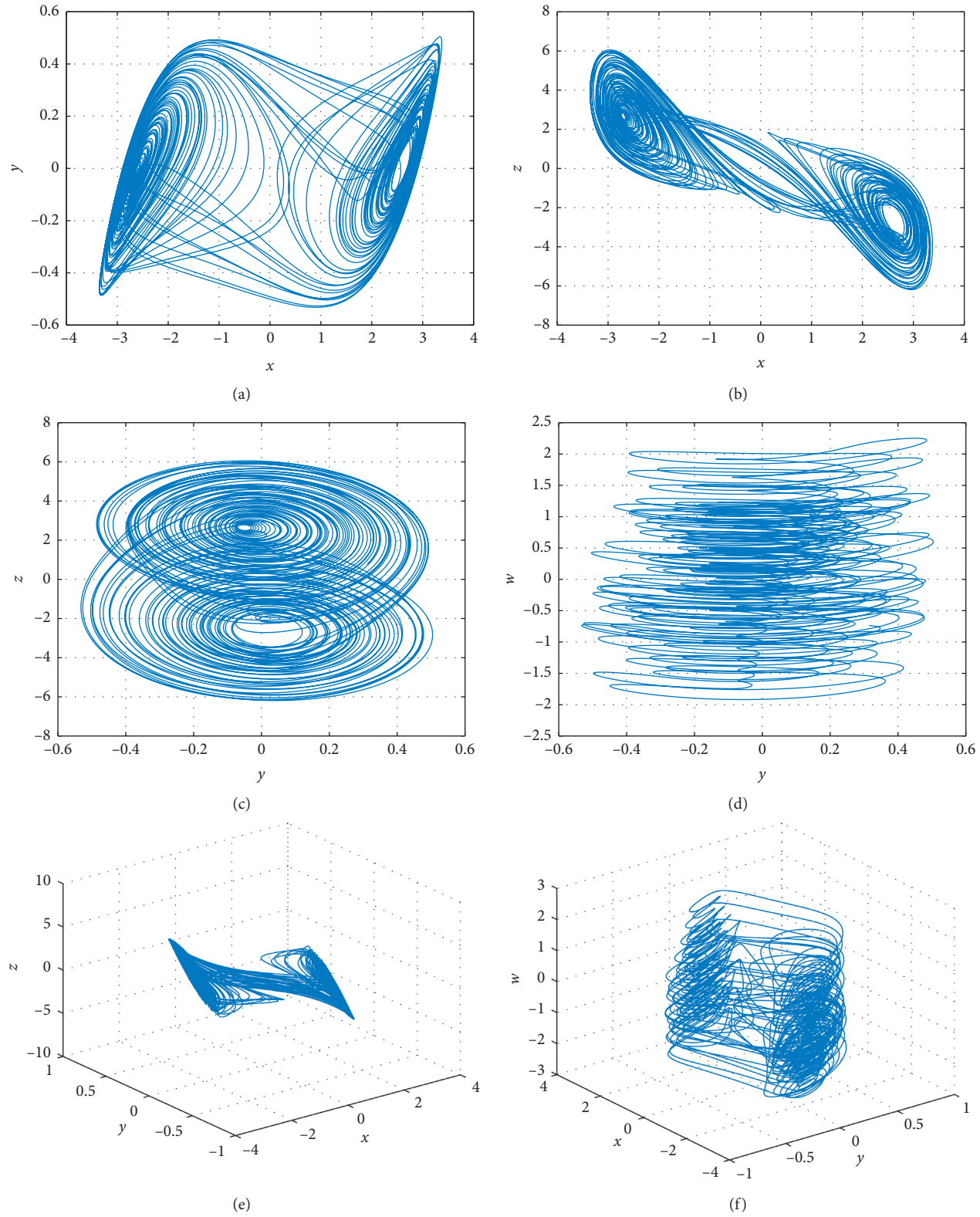


Figure 1: Continued.

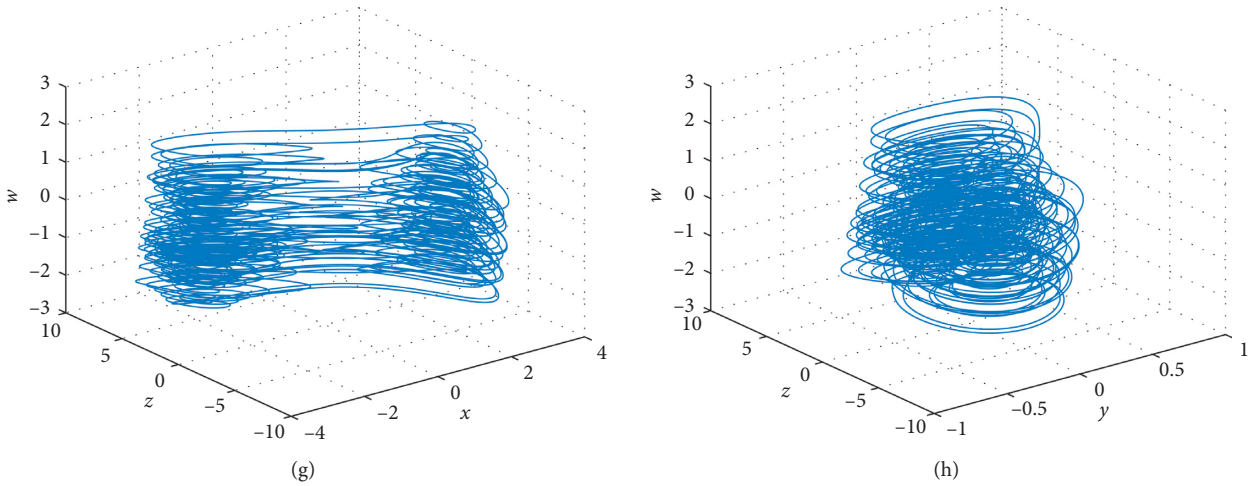


Figure 1: The phase diagram of the proposed hyperchaotic system: (a) x-y phase diagram; (b) x-z phase diagram; (c) y-z phase diagram; (d) y-w phase diagram; (e) x-y-z phase diagram; (f) x-y-w phase diagram; (g) x-z-w phase diagram; (h) y-z-w phase diagram.

$$v_{k+1} = v_k + c1 \times r1 \times (p\text{best}_k - p_k) + c2 \times r2 \times (g\text{best} - p_k), \tag{2}$$

$$p_{k+1} = v_{k+1} + p_k, \tag{3}$$

where p denotes the position of the particle and v denotes the velocity of particles, $c1$ and $c2$ are the constant learning factor, $r1$ and $r2$ are constant values between [0, 1], pbest is the best position value for each particle, and gbest is the best value for the particle swarm.

The process for implementing PSO is shown in the following algorithm (Algorithm 1).

2.3. Cellular Automata (CA). Cellular automata are not only a simple biological computing model but also a decentralized spatially extended system, which includes a large number of cells with local connectivity. CA has the potential to perform complex computation and can simulate the behavior of complex systems in nature. CA is characterized by simple rules and high efficiency. Due to its complex behavior and reliable evolutionary rules, a secure encryption system can be designed based on CA.

CA is a discrete mathematical model. In other words, CA can obtain discrete outputs through discrete inputs, and its basic cell is a simple structure that evolves in discrete time and space. CA can represent the sequential behavior of several cells connected to each other. These cells are arranged in a particular regular way, and each cell has a set of finite possible values. The state of each cell in a CA is related to the state of the adjacent cell. Thus, each cell of CA can evolve over time depending on the type of rule used.

In the one-dimensional CA, each cell has two neighborhoods, and each neighborhood has two values, namely, 0 and 1. Therefore, three cells adjacent to each other have eight possible binary states. In this paper, we use a 2-state, 3-neighborhood CA, where each cell can map the binary number of three bits to the binary number of one bit. The transition function for a 2-state, 3-neighborhood CA cell can be described by the following equation:

$$S_i^{t+1} = f\left(S_{i-1}^t, S_i^t, S_{i+1}^t\right), \tag{4}$$

where f is a Boolean function associated with each CA rule, i is the position of a cell, t is the t th time step, and $S_i^t$is the output state. If the i-th cell is to be updated, it must depend on the current state of itself and its left and right cells. Thus, the total number of rules available for CA is 256.

In addition, Wolfram encodes the set of local rules for the temporal evolution of one-dimensional CA [61]. Table 1 gives an example of Wolfram's numbering scheme for the Rule 30 of CA. The rule numbers represent the decimal value of the rule output. For example, if the mapping rule satisfies the equation (5), the binary number 00011110 is the binary form of 30:

$$\begin{cases} f(111) = 0, \\ f(110) = 0, \\ f(101) = 0, \\ f(100) = 1, \\ f(011) = 1, \\ f(010) = 1, \\ f(001) = 1, \\ f(000) = 0. \end{cases} \tag{5}$$

Therefore, the CA is called Rule 30 CA. In Boolean form, Rule 30 can be written as follows:

$$S_i^{t+1} = S_{i-1}^t \text{xor}\left[S_i^t \text{ or } S_{i+1}^t\right]. \tag{6}$$

In the image encryption system in this paper, there are 8 rules of CA used, as shown in Table 2.

Step 1: initialization process.
  Set constant values to $c1$, $c2$, $r1$, and $r2$;
  Produce the initial particle swarm;
  Randomly initialize the position and velocity of each particle.
Step 2: optimization process.
  Calculate the fitness value of each particle;
  Find the pbest;
  Find the gbest;
  If (Meet the termination condition)
    Go to Step 3
  else
    Update the velocity of each particle by equation (2);
    Update the position of each particle by equation (3);
    Go to Step 2
Step3: terminate algorithm.

Algorithm 1

Table 1: Representation of Boolean symmetry Rule 30 with radius 1.

Number	7	6	5	4	3	2	1	0
$S_{i-1}^t S_i^t S_{i+1}^t$	111	110	101	100	011	010	001	000
S_i^{t+1}	0	0	0	1	1	1	1	0

Table 2: Boolean expression for each CA rule.

R_i	Rule numbers	Boolean equation
1	30	$S_i^{t+1} = S_{i-1}^t \text{xor} [S_i^t \text{ or } S_{i+1}^t]$
2	90	$S_i^{t+1} = S_{i-1}^t \text{xor} S_{i+1}^t$
3	150	$S_i^{t+1} = S_{i-1}^t \text{xor} S_i^t \text{xor} S_{i+1}^t$
4	153	$S_i^{t+1} = S_i^t \text{xnor} S_{i+1}^t$
5	165	$S_i^{t+1} = S_{i-1}^t \text{xnor} S_{i+1}^t$
6	86	$S_i^{t+1} = [S_{i-1}^t \text{nor} S_i^t] \text{xor} [\text{not} S_i^t]$
7	105	$S_i^{t+1} = \text{not} [S_{i-1}^t \text{xor} S_i^t \text{xor} S_{i+1}^t]$
8	101	$S_i^{t+1} = [S_{i-1}^t \text{nor} S_{i+1}^t] \text{ or } [(S_i^t \text{xor} S_{i+1}^t) \text{ and } S_{i-1}^t]$

If all CA cells follow the same rules, the CA is called a unified CA; otherwise, the CA is not uniform. Moreover, if many cells are adjacent to each other, CA is considered to have periodic boundary conditions. Otherwise, the CA is referred to as a borderless CA [62].

3. The Proposed Cryptosystem

3.1. Generation of Initial Values of the Hyperchaotic System. In this paper, in order to enhance the correlation between the encryption system and the plain image and improve the encryption system's resistance to plaintext attack, the SHA-256 hash function of the original image is used to calculate the initial values of the hyperchaotic system. In cases where only 1 bit is different, the hash values of the two images are significantly different, which helps to increase the sensitivity of the encryption system to the secret key. Firstly, before encrypting the plain image, calculate the 256 bit hash value of the plain image and set it to the secret key K. Then, K is divided into 32 8 bit blocks, and each block is converted to a decimal digit. Next, the initial values of the chaotic system are calculated by the following steps.

Step 1: convert K into 32 decimal numbers $k_1, k_2, \ldots, k_{32}$, and then obtain m_1, m_2, m_3, and m_4 by equation (7):

$$\begin{cases} m_1 = \dfrac{k_1 + k_2 + k_3 + \cdots\cdots + k_8}{256}, \\ m_2 = \dfrac{k_9 + k_{10} + k_{11} + \cdots\cdots + k_{16}}{256}, \\ m_3 = \dfrac{k_{17} + k_{18} + k_{19} + \cdots\cdots + k_{24}}{256}, \\ m_4 = \dfrac{k_{25} + k_{26} + k_{27} + \cdots\cdots + k_{32}}{256}. \end{cases} \tag{7}$$

Step 2: utilize $m_1, m_2, m_3,$ and m_4 to calculate $x_0, y_0, z_0,$ and w_0 as the initial values of the chaotic system by the following equation:

$$\begin{cases} x_0 = t_1 + abs(m_1) - \text{fix}(m_1), \\ x_0 = t_1 + abs(m_1) - \text{fix}(m_1), \\ z_0 = t_3 + abs(m_3) - \text{fix}(m_3), \\ w_0 = t_4 + abs(m_4) - \text{fix}(m_4), \end{cases} \tag{8}$$

where $t_1, t_2, t_3,$ and t_4 are parts of secret keys, $\text{abs}(x)$ is the absolute value of x, and $\text{fix}(x)$ return the integer value of x.

3.2. Encryption Process. The population is the set of all images produced by a round of encryption process in PSO, so a picture is an individual. The flow diagram of the encryption system is shown in Figure 2.

The detailed steps of the encryption process are as follows:

Step 1: let P be the plain image which size is $N \times N$. Utilize the initial values$x_0, y_0, z_0,$ and $w_0$of the hyperchaotic system to iterate the hyperchaotic system for $T + L$ times, where $L = N \times \text{N}$. To avoid the effect of transient, we discard the previous T values. Finally, we get four chaotic sequences X, Y, Z, and W, each of which consists of L elements. Then, according to equation (9), four sequences $X_1, Y_1, Z_1,$ and $W_1$are obtained:

$$\begin{cases} X_1(i) = \text{mod}\left(X(i) \times 10^8, 1\right), \\ Y_1(i) = \text{mod}\left(abs(Y(i) - Z(i)) \times 10^8, 1\right), \\ Z_1(i) = \text{mod}\left(abs(X(i) + Y(i) + Z(i)) \times 10^8, 1\right), \\ W_1(i) = \text{mod}\left(abs(W(i)) \times 10^8, 3\right). \end{cases} \tag{9}$$

Step 2: the initialization of PSO.

Step 2.1: set the population size popSize and learning factors c1 and c2. The chaotic sequence X_1 is used to initialize each individual in the population, and the process of generating individuals is shown in Figure 3.
Step 2.2: initialize the velocity and position of each individual according to the following equations:

$$v = \frac{\text{floor}(y)}{N} + 1, \tag{10}$$

$$p = \text{fix}\left(\text{mod}\left(y \times 10^5, N^* N\right)\right). \tag{11}$$

where y is an element in the sequence Y_1, v denotes the velocity of the individual, and p denotes the position of the individual.
Step 2.3: set pbest and gbest. In this paper, the correlation coefficient between adjacent pixels of the image is taken as the fitness function of the optimization process. Therefore, the smaller the fitness function value, the more individuals will be retained into the next generation.

Step 3: perform the optimization process.

Step 3.1: set $s = 1$, where s is the number of rounds in the optimization process.
Step 3.2: update the velocity and position of each generated individual according to equations (2) and (3), where $r1$ and $r2$ use elements from chaotic sequences $Y_1$and W_1, respectively.
Step 3.3: transform the rows and columns of a two-dimensional image into one-dimensional data, and the pixel value of the image is diffused by using rule 150 in cellular automata technology, as shown in the following equation.

$$e_i = e_{i-1} \oplus e_i \oplus e_{i+1}. \tag{12}$$

where the first pixel and the last pixel are updated according to the following equation:

$$\begin{cases} e_1 = e_{M^* N} \oplus e_1 \oplus e_2, \\ e_{N^* N} = e_{N^* N-1} \oplus e_{N^* N} \oplus e_1. \end{cases} \tag{13}$$

Step 3.4: reconstruct the rows and columns of the one-dimensional data to restore the two-dimensional image and then calculate the fitness of each individual and update the pbest and gbest.
Step 3.5: set $s = s+1$, then loop executes Step 3.2 to Step 3.5 S times, and the cipher image C is obtained.

3.3. Decryption Process. The reverse of the encryption process is the decryption process. Before decrypting, the sender sends the secret keys safely to the receiver. Secret keys include the 256-bit hash value K, the parameters $(t_1, t_2, t_3,$ and $t_4)$, discarding T numbers of chaotic sequences, the total number S of particle swarm optimization algorithm rounds, and the velocity V and position P of the best individual. After the recipient obtains the secret keys, the decryption of cipher image only requires the inverse process of diffusion. The detailed decryption process is as follows:

Step 1: the initial value $(x_0, y_0, z_0,$ and $w_0)$of chaotic system is obtained, according to Section 3.2.

Step 2: iterate the hyperchaotic system for $T + L$ times, where $L = \text{N} \times \text{N}$. To avoid the transient effect, we discard the previous T values. Then, four new hyperchaotic sequences are obtained by equation (9).

Step 3: set pbest and gbest, which is generally a larger value.

Step 4: set $s = 1$, and s represents the number of rounds in the current optimization process.

Step 4.1: carry out the reverse diffusion process of cellular automata according to equations (12) and (13).
Step 4.2: update the velocity and position of each generated individual according to equations (14) and (15):

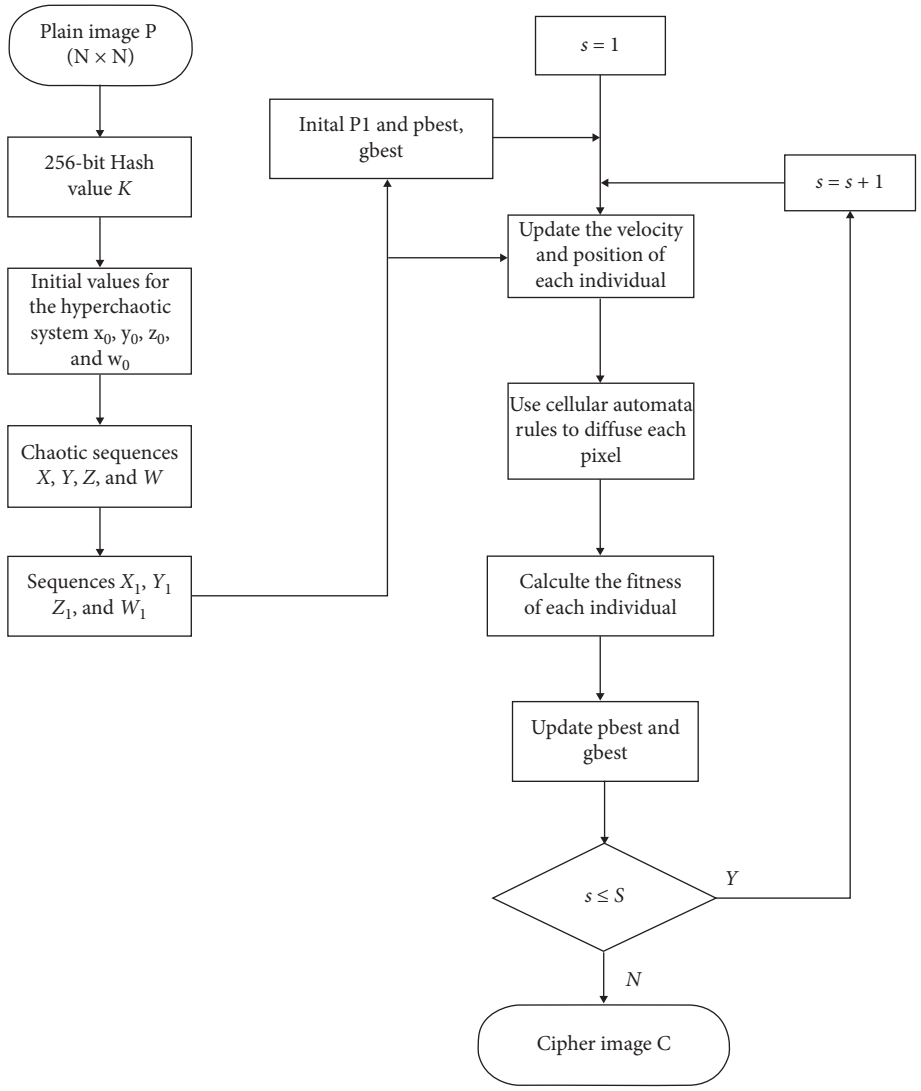


Figure 2: The flow diagram of the encryption system.

$$v_k = v_{k+1} - c1 \times r1 \times (p\text{best}_k - p_k) - c2 \times r2 \times (g\text{best} - p_k), \quad (14)$$

$$p_k = p_{k+1} - v_{k+1}. \quad (15)$$

Step 4.3: set $s = s+1$, and then loop executes Step 4.1 to Step 4.3 S times. Finally, the plain image P is recovered.

4. Simulation Results and Security Analysis

4.1. Simulation Results. To demonstrate the effectiveness of the proposed system given above, the simulation process is carried out on the MATLAB software platform. For color images, the system can encrypt the image data of *R*, G and B channels, respectively. After encryption, they are recombined to obtain color cipher images. In addition, the encryption system can effectively encrypt images of various sizes. In other words, the encryption system has no limit on the size of the image. Take the plain image of Lena for encryption and decryption, as shown in Figure 4.

4.2. Statistical Analysis

4.2.1. Histogram. Histograms can be used to count the number of values with a specific gray value. If a high security image encryption system is used, cipher images with uniform histograms can be obtained.

As shown in Figure 5, compared with the histogram of plain image, cipher image has the characteristic of uniform distribution of pixel values. The results of gray histogram analysis show that the proposed encryption system can resist statistical attacks.

4.2.2. Correlation of Adjacent Pixels. The correlation coefficient between image pixels can represent the statistical relationship between neighborhood pixels. An ideal encryption system should be able to break high correlation.

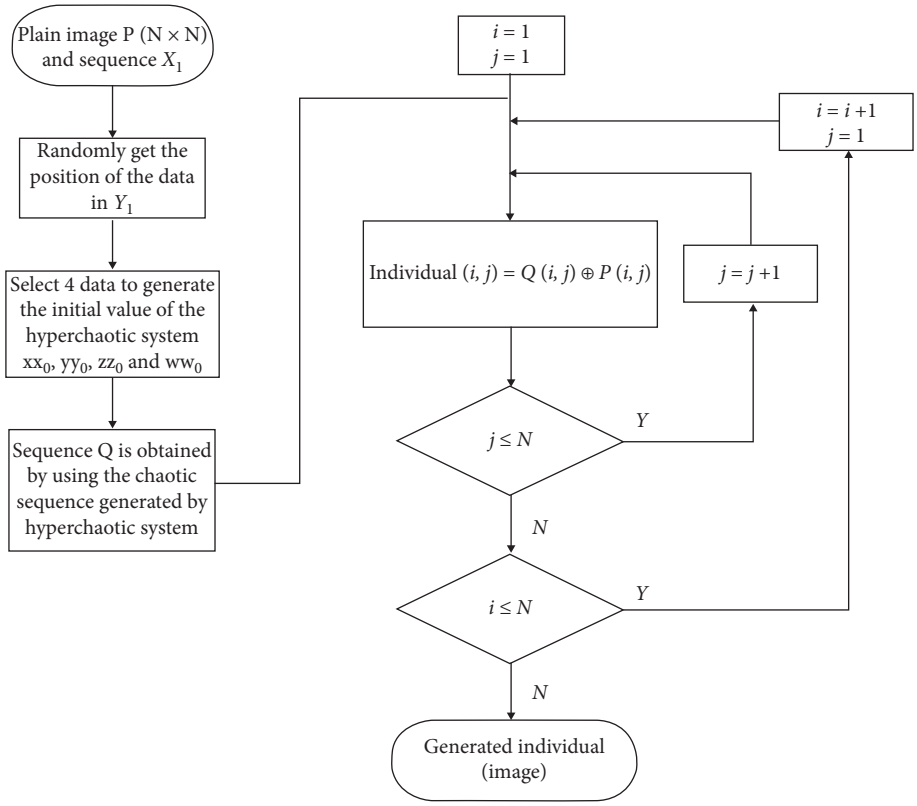


Figure 3: The flow diagram of generating individual.

(a) (b)

Figure 4: Continued.

(c)

Figure 4: The simulation results: (a) the plain image; (b) the cipher image; (c) the recovered image.

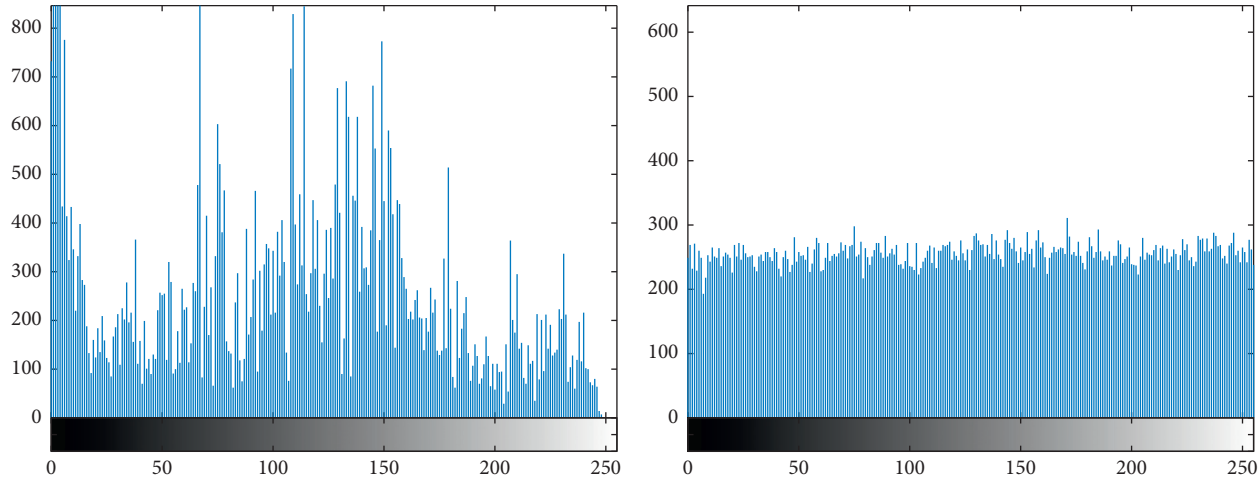


Figure 5: Histogram analysis: the histogram of (a) plain image of Lena and (b) cipher images of Lena.

between adjacent pixels of the plain image.

From the tested images, a number of pixels were randomly selected to calculate the correlation coefficients in three different directions (horizontal, vertical, and diagonal). The correlation coefficients are calculated by the following formulae:

$$r_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)D(y)}}, \tag{16}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{17}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)^2, \tag{18}$$

$$\operatorname{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)\left(y_i - E(y)\right), \tag{19}$$

where x and y are two neighborhood pixel values in the image, respectively.

Figure 6 shows the correlation distribution in the three directions. As shown in Table 3, by comparing the correlation coefficients in the three directions, it can be concluded

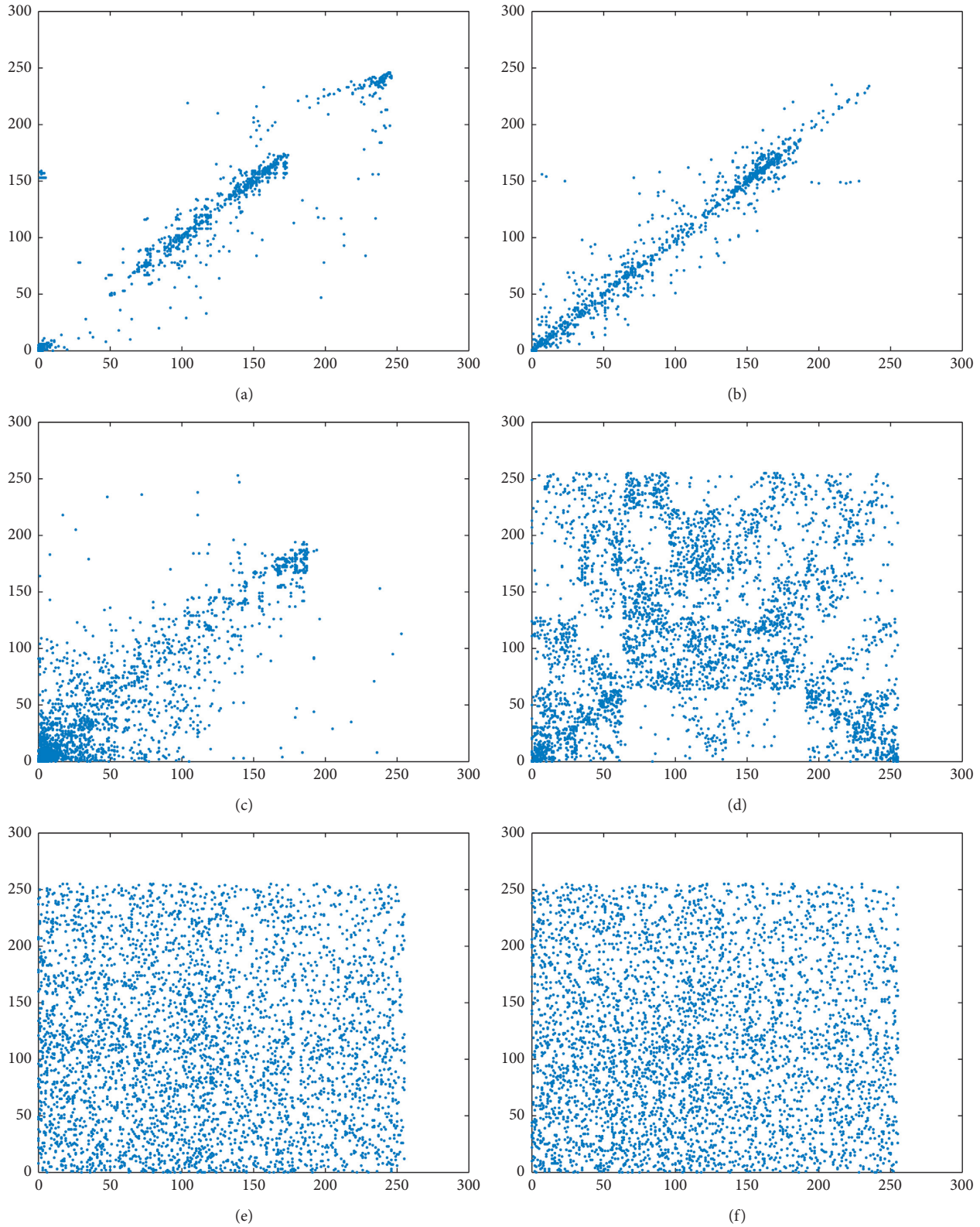


Figure 6: Correlation analysis: (a) horizontal of plain image; (b) vertical of plain image; (c) diagonal of plain image; (d) horizontal of cipher image; (e) vertical of cipher image; (f) diagonal of cipher image.

Table 3: Correlation coefficients of plain image and cipher image.

Image	Direction		
	Horizontal	Vertical	Diagonal
Plain image (Lena)	0.9567	0.9611	0.9169
Cipher image (Lena)	0.0053	−0.0089	0.0126

that the correlation between adjacent pixels of cipher image is much lower than that of plain image. Therefore, the proposed encryption system has a good performance in resisting statistical analysis attacks.

4.3. Sensitivity Analysis

4.3.1. Differential Attack. Image encryption schemes generally require that the encrypted image should be vastly different from the original plain image. The proposed encryption system ensures that two cipher images are completely different, even if only one bit of data is different between the plain images. Differential attack means to break the encryption algorithm by comparing and analyzing the corresponding cipher images of two plain images with slight differences. Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are two standards used to test the ability of encryption algorithm to resist differential attack. The calculation formula is shown in the following equations:

$$\mathrm{NPCR} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M \times N} \times 100\%, \tag{20}$$

$$\mathrm{UACI} = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\frac{\left|C'(i,j) - C(i,j)\right|/255)}{M \times N \times 100\%,}\right. \tag{21}$$

where M and N represent the number of rows and columns of the image, C is the encrypted image of the plain image, and C' is the encrypted image corresponding to the plain image after a slight change. The definition of D is shown as follows:

$$D(i,j) = \begin{cases} 1, & \text{if } C(i,j) \neq C'(i,j), \\ 0, & \text{if } C(i,j) = C'(i,j). \end{cases} \tag{22}$$

The results in Table 4 show that, after the plain image is processed by the encryption system, the NPCR and UACI values of the proposed encryption system are 99.6352% and 33.5614%, respectively, which are close to the ideal values. Therefore, our proposed system is more effective in resisting differential attacks.

4.3.2. Key Sensitivity. For key sensitivity, if the key is slightly modified, a secure encryption system should get a completely different cipher image in the encryption process. A noisy image without any information about plain image should be obtained in the decryption process.

Take image Lena as an example, in the encryption process, we encrypt with the original key and then encrypt with the modified key. As shown in Figure 7, a cipher image is obtained with the original key; however, we encrypt the plain image again by using the modified key to obtain a completely different cipher image. It is obvious that a slight modification of the key can make a huge difference in the cipher image. In the decryption process, the cipher image of Lena is decrypted by the original key and the modified key, respectively. As shown in Figure 8, the original key is successfully restored the original plain image, but the slightly modified key could not correctly restore the plain image.

4.4. Information Entropy.

In information theory, information entropy is used to evaluate the amount of information. The calculation method of entropy is described as the following formula:

$$H = -\sum_{i=0}^{2^N-1} p(m_i)\log_2 p(m_i), \tag{23}$$

where N is the number of bits for m_i and $p(m_i)$ is the probability of m_i. Table 5 lists the values of the tested image information entropy, and the calculated results are close to the ideal theoretical value of 8. Therefore, the proposed encryption system can achieve a better randomness effect on cipher images by encrypting plain images.

4.5. Encryption and Decryption Efficiencies.

Encryption and decryption efficiency refers to the running speed of the system. The hardware environment used in this paper is the computer Windows 10 64 bit operating system, the Intel Core I5-8300h processor @2.30ghz and 8 GB of memory. The software environment is Matlab (R2017a). In the same hardware and software environment, the running speed of encryption system plays a crucial role in satisfying the real-time performance of network transmission. In this paper, the PSO is simple in coding and genetic operation and has the advantages of fewer parameters, easier implementation, and faster optimization speed. Therefore, the speed and performance of the encryption system are guaranteed. When the plain image is encrypted, cipher image can be obtained quickly.

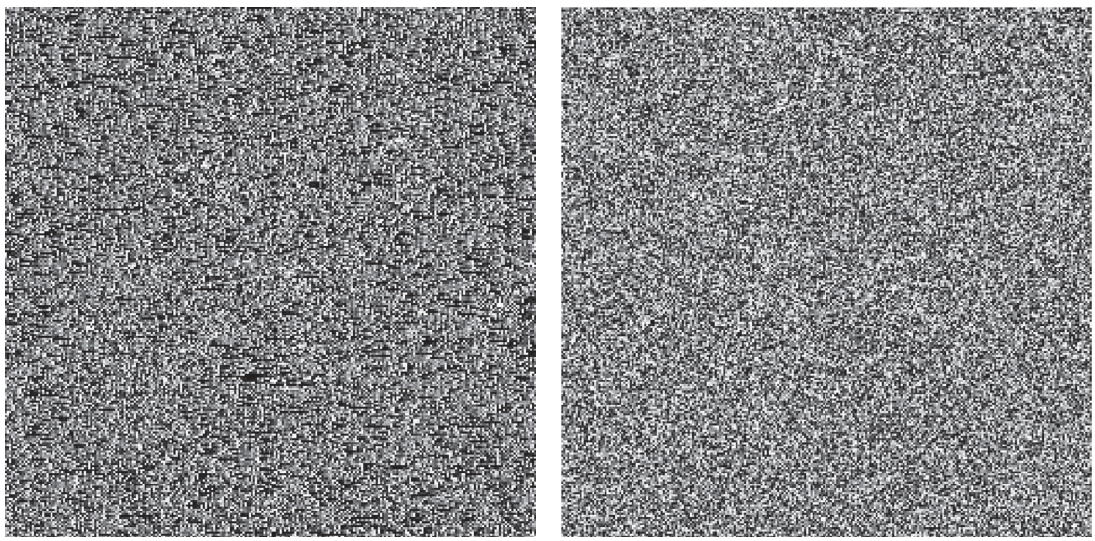
4.6. Key Space Analysis.

The key of the encryption system proposed in this paper includes the given initial key and the hash value of the plain image. The initial values are double-precision stored inside the computer.

Therefore, the key space is $(10^{16})^8 \times 2^{128} \approx 2^{512} > 2^{100}$. For a secure encryption system, the key space should be greater than 2^{100}. Obviously, it is not feasible for malicious attackers to exploit exhaustive attacks to break the encryption system.

Table 4: NPCR and UACI of image Lena.

Image	NPCR (%)	UACI (%)
Lena	99.6352	33.5614



(a) (b)

Figure 7: Sensitivity test of encryption key: (a) encrypted image with the original secret key; (b) encrypted image with the modified secret key.

(a) (b)

Figure 8: Sensitivity test of decryption key: (a) decrypted image with the correct secret key; (b) encrypted image with the modified secret key.

Table 5: Information entropy.

Image	Plain image	Cipher image
Lena	7.5827	7.9987

4.7. Encryption Performance Comparison. Taking image Lena as an example, Table 6 lists the performance comparison between the encryption system proposed in this paper and other encryption systems. We have compared the correlation coefficients, the NPCR and UACI values, and information entropy of the proposed encryption system and

Table 6: Encryption performance comparison.

Encryption scheme	Direction			NPCR (%)	UACI (%)	Information entropy
	Horizontal	Vertical	Diagonal			
Ours	0.0053	−0.0089	0.0126	99.6352	33.5614	7.9987
Ref. [31]	−0.0574	−0.0035	0.0578	99.5839	33.3756	7.9989
Ref. [44]	−0.0016	−0.0033	0.0130	99.6125	33.5375	7.9971
Ref. [48]	0.0015	0.0043	0.0023	99.6881	37.5600	7.9877

other systems. Therefore, we can conclude that our encryption system has better comprehensive performance than the other three encryption methods.

5. Conclusion

In this paper, a new hyperchaotic image encryption system based on PSO and CA is proposed. This system increases the complexity and diversity of the population in the process of evolution, increases the key space of the encryption system, reduces the possibility of the loss of high-quality population, and enhances the security of the encryption system. In security analysis, we compared the security of our system with other encryption systems' security performance in the relevant literature and verified the security of the image encryption system through histogram analysis, difference analysis, correlation analysis, statistical analysis, key performance analysis, etc. The performance analysis demonstrate that the proposed system has better encryption effect and higher resistance against plaintext attack and statistical attack in comparison to other hyperchaotic image encryption systems.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61971185) and Natural Science Foundation of Hunan Province (2020JJ4218).

References

[1] M. S. Baptista, "Cryptography with chaos," *Physics Letters A*, vol. 240, no. 1-2, pp. 50–54, 1998.

[2] R. Schmitz, "Use of chaotic dynamical systems in cryptography," *Journal of the Franklin Institute*, vol. 338, no. 4, pp. 429–441, 2001.

[3] M. Zhu, C. Wang, Q. Deng, and Q. Hong, "Locally active memristor with three coexisting pinched hysteresis loops and its emulator circuit," *International Journal of Bifurcation and Chaos*, vol. 30, no. 13, Article ID 2050184, 2020.

[4] H. Lin, C. Wang, Q. Hong, and Y. Sun, "A multi-stable memristor and its application in a neural network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3472–3476, 2020.

[5] Q. Deng, C. Wang, and L. Yang, "Four-wing hidden attractors with one stable equilibrium point," *International Journal of Bifurcation and Chaos*, vol. 30, no. 06, Article ID 2050086, 2020.

[6] H. Lin, C. Wang, W. Yao, and Y. Tan, "Chaotic dynamics in a neural network with different types of external stimuli," *Communications in Nonlinear Science and Numerical Simulation*, vol. 90, Article ID 105390, 2020.

[7] C. Wang, H. Xia, and L. Zhou, "A memristive hyperchaotic multiscroll jerk system with controllable scroll numbers," *International Journal of Bifurcation and Chaos*, vol. 27, no. 06, Article ID 1750091, 2017.

[8] H. Lin, C. Wang, Y. Sun, and W. Yao, "Firing multistability in a locally active memristive neuron model," *Nonlinear Dynamics*, vol. 100, no. 4, pp. 3667–3683, 2020.

[9] Y. M. Tan and C. H. Wang, "A simple locally active memristor and its application in HR neurons," *Chaos*, vol. 30, no. 5, Article ID 053118, 2020.

[10] F. Yu, L. Liu, H. Shen et al., "Multistability analysis, coexisting multiple attractors, and FPGA implementation of yu-wang four-wing chaotic system," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–16, Article ID 7530976, 2020.

[11] F. Yu, S. Qian, X. Chen et al., "A new 4D four-wing memristive hyperchaotic system: dynamical analysis, electronic circuit design, shape synchronization and secure communication," *International Journal of Bifurcation and Chaos*, vol. 30, no. 10, Article ID 2050147, 2020.

[12] R. Wu and C. Wang, "A new simple chaotic circuit based on memristor," *International Journal of Bifurcation and Chaos*, vol. 26, no. 9, Article ID 1650145, 2016.

[13] C. Wang, H. Xia, and L. Zhou, "Implementation of a new memristor-based multiscroll hyperchaotic system," *Pramana-Journal of Physics*, vol. 88, no. 2, pp. 1–7, 2017.

[14] W. Yao, C. H. Wang, Y. C. Sun, C. Zhou, and H. R. Lin, "Exponential multistability of memristive Cohen-Grossberg neural networks with stochastic parameter perturbations," *Applied Mathematics and Computation*, vol. 386, Article ID 125483, 2020.

[15] F. Yu, S. Qian, X. Chen et al., "Chaos-based engineering applications with a 6D memristive multistable hyperchaotic system and a 2D SF-SIMM hyperchaotic map," *Complexity*, vol. 2021, Article ID 6683284, 2021.

[16] X. Zhang and C. Wang, "Multiscroll hyperchaotic system with hidden attractors and its circuit implementation," *International Journal of Bifurcation and Chaos*, vol. 29, no. 9, 2019.

[17] Y. Li, Z. Li, M. Ma et al., "Generation of grid multi-wing chaotic attractors and its application in video secure communication system," *Multimedia Tools and Applications*, vol. 79, no. 1, pp. 29161–29177, 2020.

[18] C. Zhou, F. Xie, and Z. Li, "Complex bursting patterns and fast-slow analysis in a smallest chemical reaction system with two slow parametric excitations," *Chaos, Solitons & Fractals*, vol. 137, 2020.
[19] O. E. Rossler, "An equation for hyperchaos," *Physics Letters A*, vol. 71, no. 2-3, pp. 155–157, 1979.
[20] M. Khan, "A novel image encryption scheme based on multiple chaotic S-boxes," *Nonlinear Dynamics*, vol. 82, no. 1-2, pp. 527–533, 2015.
[21] Y. Liu, X. Tong, and J. Ma, "Image encryption algorithm based on hyper-chaotic system and dynamic S-box," *Multimedia Tools and Applications*, vol. 75, no. 13, pp. 7739–7759, 2016.
[22] P. Devaraj and C. Kavitha, "An image encryption scheme using dynamic S-boxes," *Nonlinear Dynamics*, vol. 86, no. 2, pp. 927–940, 2016.
[23] Y. Zhang, "The unified image encryption algorithm based on chaos and cubic S-Box," *Information Sciences*, vol. 450, pp. 361–377, 2018.
[24] X. Zhang, W. Nie, Y. Ma, and Q. Tian, "Cryptanalysis and improvement of an image encryption algorithm based on hyper-chaotic system and dynamic S-box," *Multimedia Tools and Applications*, vol. 76, no. 14, pp. 15641–15659, 2017.
[25] Ü. Çavuşoğlu, S. Kaçar, I. Pehlivan et al., "Secure image encryption algorithm design using a novel chaos based S-box," *Chaos, Solitons and Fractals*, vol. 95, pp. 92–101, 2017.
[26] C. Zhu, G. Wang, and K. Sun, "Cryptanalysis and improvement on an image encryption algorithm design using a novel chaos based S-box," *Symmetry*, vol. 10, no. 9, p. 399, 2018.
[27] X. Wang, Üu Çavuşoğlu, S. Kacar et al., "S-box based image encryption application using a chaotic system without equilibrium," *Applied Sciences*, vol. 9, no. 4, p. 781, 2019.
[28] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," *Optics and Lasers in Engineering*, vol. 78, pp. 17–25, 2016.
[29] W. Zhang, H. Yu, Y.-l. Zhao, and Z.-l. Zhu, "Image encryption based on three-dimensional bit matrix permutation," *Signal Processing*, vol. 118, pp. 36–50, 2016.
[30] Y. Li, C. Wang, and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Optics and Lasers in Engineering*, vol. 90, pp. 238–246, 2017.
[31] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Optics Communications*, vol. 284, no. 16-17, pp. 3895–3903, 2011.
[32] J. Liu, D. Yang, H. Zhou, and S. Chen, "A digital image encryption algorithm based on bit-planes and an improved logistic map," *Multimedia Tools and Applications*, vol. 77, no. 8, pp. 10217–10233, 2018.
[33] J. Wu, X. Liao, and B. Yang, "Cryptanalysis and enhancements of image encryption based on three-dimensional bit matrix permutation," *Signal Processing*, vol. 142, pp. 292–300, 2018.
[34] A. Jain and N. Rajpal, "A robust image encryption algorithm resistant to attacks using DNA and chaotic logistic maps," *Multimedia Tools and Applications*, vol. 75, no. 10, pp. 5455–5472, 2016.
[35] S. Zhang and T. Gao, "An image encryption scheme based on DNA coding and permutation of hyper-image," *Multimedia Tools and Applications*, vol. 75, no. 24, pp. 17157–17170, 2016.
[36] T. Hu, Y. Liu, L.-H. Gong, and C.-J. Ouyang, "An image encryption scheme combining chaos with cycle operation for DNA sequences," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 51–66, 2017.
[37] J. Wu, X. Liao, and B. Yang, "Image encryption using 2D Hénon-Sine map and DNA approach," *Signal Processing*, vol. 153, pp. 11–23, 2018.
[38] X. Chai, X. Fu, Z. Gan, Y. Lu, and Y. Chen, "A color image cryptosystem based on dynamic DNA encryption and chaos," *Signal Processing*, vol. 155, pp. 44–62, 2019.
[39] S. Kayalvizhi and S. Malarvizhi, "A novel encrypted compressive sensing of images based on fractional order hyper chaotic Chen system and DNA operations," *Multimedia Tools and Applications*, vol. 79, no. 5-6, pp. 3957–3974, 2020.
[40] S. Wang, C. Wang, and C. Xu, "An image encryption algorithm based on a hidden attractor chaos system and the Knuth-Durstenfeld algorithm," *Optics and Lasers in Engineering*, vol. 128, p. 105995, 2020.
[41] Y. Wan, S. Gu, and B. Du, "A new image encryption algorithm based on composite chaos and hyperchaos combined with DNA coding," *Entropy*, vol. 22, no. 2, p. 171, 2020.
[42] X. Wang and D. Xu, "A novel image encryption scheme using chaos and Langton's Ant cellular automaton," *Nonlinear Dynamics*, vol. 79, no. 4, pp. 2449–2456, 2015.
[43] R. Enayatifar, H. J. Sadaei, A. H. Abdullah, M. Lee, and I. F. Isnin, "A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata," *Optics and Lasers in Engineering*, vol. 71, pp. 33–41, 2015.
[44] X. Chai, Z. Gan, K. Yang, Y. Chen, and X. Liu, "An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations," *Signal Processing: Image Communication*, vol. 52, pp. 6–19, 2017.
[45] A. Y. Niyat, M. H. Moattar, and M. N. Torshiz, "Color image encryption based on hybrid hyper-chaotic system and cellular automata," *Optics and Lasers in Engineering*, vol. 90, pp. 225–237, 2017.
[46] M. Li, D. Lu, W. Wen, H. Ren, and Y. Zhang, "Cryptanalyzing a color image encryption scheme based on hybrid hyper-chaotic system and cellular automata," *IEEE Access*, vol. 6, pp. 47102–47111, 2018.
[47] X. Di, L. Wang, T. Xiang et al., "Multi-focus image fusion and robust encryption algorithm based on compressive sensing," *Optics & Laser Technology*, vol. 91, pp. 212–225, 2017.
[48] B. Mondal, S. Singh, and P. Kumar, "A secure image encryption scheme based on cellular automata and chaotic skew tent map," *Journal of Information Security and Applications*, vol. 45, pp. 117–130, 2019.
[49] Q. Yin and C. Wang, "A new chaotic image encryption scheme using breadth-first search and dynamic diffusion," *International Journal of Bifurcation and Chaos*, vol. 28, no. 4, 2018.
[50] M. Zhou and C. Wang, "A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks," *Signal Processing*, vol. 171, 2020.
[51] C. Xu, J. Sun, and C. Wang, "An image encryption algorithm based on random walk and hyperchaotic systems," *International Journal of Bifurcation and Chaos*, vol. 30, no. 4, Article ID 2050060, 2020.
[52] G. Cheng, C. Wang, and C. Xu, "A novel hyper-chaotic image encryption scheme based on quantum genetic algorithm and compressive sensing," *Multimedia Tools and Applications*, vol. 79, no. 39-40, pp. 29243–29263, 2020.
[53] G. Ye, K. Jiao, H. Wu, C. Pan, and X. Huang, "An asymmetric image encryption algorithm based on a fractional-order chaotic system and the RSA public-key cryptosystem," *International Journal of Bifurcation and Chaos*, vol. 30, no. 15, Article ID 2050233, 2020.

[54] G. Ye, C. Pan, Y. Dong, Y. Shi, and X. Huang, "Image encryption and hiding algorithm based on compressive sensing and random numbers insertion," *Signal Processing*, vol. 172, Article ID 107563, 2020.
[55] G. Ye, C. Pan, X. Huang, and Q. Mei, "An efficient pixel-level chaotic image encryption algorithm," *Nonlinear Dynamics*, vol. 94, no. 1, pp. 745–756, 2018.
[56] C. Xu, J. Sun, and C. Wang, "A novel image encryption algorithm based on bit-plane matrix rotation and hyper chaotic systems," *Multimedia Tools and Applications*, vol. 79, no. 9-10, pp. 5573–5593, 2020.
[57] J. Sun, "Protecting compressive ghost imaging with hyperchaotic system and DNA encoding," *Complexity*, vol. 2020, no. 11, p. 13, Article ID 8815315, 2020.
[58] M. Ahmad, M. Z. Alam, Z. Umayya, S. Khan, and F. Ahmad, "An image encryption approach using particle swarm optimization and chaotic map," *International Journal of Information Technology*, vol. 10, no. 3, pp. 247–255, 2018.
[59] P. C. Rech and H. A. Albuquerque, "A hyperchaotic chua system," *International Journal of Bifurcation and Chaos*, vol. 19, no. 11, pp. 3823–3828, 2009.
[60] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Mhs95 Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.
[61] S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of Modern Physics*, vol. 55, no. 3, pp. 601–644, 1983.
[62] S. M. Hosseini, H. Karimi, and M. V. Jahan, "Generating pseudo-random numbers by combining two systems with complex behaviors," *Journal of Information Security and Applications*, vol. 19, no. 2, pp. 149–162, 2014.