Hindawi
International Journal of Optics
Volume 2020, Article ID 8678527, 11 pages
https://doi.org/10.1155/2020/8678527

Research Article

Image Encryption Based on High-Dimensional Manifold Computing and Block Dividing Algorithm

Meng Jia

Xinxiang Univerisity, Ürümqi, China

Correspondence should be addressed to Meng Jia; jiameng@xxu.edu.cn

Received 15 September 2019; Revised 1 February 2020; Accepted 26 February 2020; Published 11 August 2020

Academic Editor: John T. Sheridan

The image encryption schemes developing more sensitive and more chaotic maps are used as a key sequence generator, such as cascade chaotic maps and high-dimension maps. High-dimension chaotic maps can generate sequences with little correlation after hundred times of iteration. In fact, the sequence is just a flow of the manifold of the chaotic system. A fast way to select sensitive flows of the dynamic systems is introduced in the paper. Combining with the divided blocks diffusion algorithm, the novel scheme yields the cipher image more randomly. Experimental results show that the sensitive flows of the high-dimension system can generate a series with better ergodicity and less correlation compared with the common flows of the same system. The encryption efficiency is enhanced by choosing the sensitive flows of the high-dimension system. The analysis proved that the novel image encryption scheme can resist all common kinds of attacks.

1. Introduction

Image encryption has been widely used in information privacy, national safety, medical image protection, and other fields where only the right person gets the exactly true message. In the recent years, many encryption theories have been used in image encryption, such as elliptic curve approach [1, 2], cellular automata theory [3], DNA encryption [4–7], secure hash algorithm [8], image compress method [9], biological operations [10], and image fusion technology, watermarking, and chaos encryption [11–15]. Among the theories mentioned above, the chaos encryption methods have been proved to be effective for its sensitivity, diffusion ability, and confusion ability. The chaos encryption approach does bifurcation analysis to the chosen nonlinear dynamic system, and then it takes advantage of the chaos nonlinear dynamic system to generate pseudorandom sequences which are used for image encryption [16, 17]. In the other way, the encryption attack methods are also developed. There are brute-force attack [18], known-plaintext attack [19], chosen-plaintext attack [20], chosen ciphertext attack [21], differential attack [22, 23], meet-in-middle attack [24–26], and so on. So more complex systems, bigger key space, and higher sensitivity methods are needed to resist different kinds of attacks.

Wu et al. introduce an encryption method based on the chaos system and the elliptic curve [1]. This method uses elliptic theory to compress the original image first and then takes advantage of the 4-dimension system to do diffusion and confusion. Although the results show that both the pixel correlation and structural similarity index metrics are good, it has poor ability for noise attack. Lan et al. combine the sharing matrix and image encryption for lossless (k, n)-secret image sharing [27]. This approach introduces the (k, n) sharing matrix producing algorithm, and it combines the sharing matrix with chaos encryption to do image encryption. The results show that it is sensitive for the initial value and has a high key space and can detect fake share. The most important is that the encrypted image has n shares, and only k shares are enough for decryption, where $k< n$. This means that even some information get lost in transmission, and we can still make the encrypted image decrypt without information loss. The disadvantages are that the sharing matrix computing costs too much time, and it cannot resist noise attack. Lee et al. proposed a parallel computing method for chaos encryption [12]. It divided the original

image into blocks first and then did chaos encryption synchronously with multithreading. This method improved the computing speed greatly. Although it made the encryption methods with the chaos system more executable, not all the software supports multithreading operation. An approach combining DNA algorithms and chaos maps was presented in reference [8]. In that paper, the Watson–Crick complementary rule was used for bit-level changing, and chaotic maps generated a sequence to choose different biological and algebraic operations. It also introduces SHA-256 to the encryption process which increases the complexity. This method can resist noise attack, but key sensitivity is not good enough. Cellular automata (CA) and chaotic map to do encryption were employed in reference [8]. The CA rule made the "0" and "1" distribution in equilibrium in the process of iteration, and the chaotic map is used to produce a sequence to shuffle pixel position and bits confusion. A network encryption method was proposed in reference [3]. It did diffusion and confusion at the same time, and it improved the computing speed. By integrating more than one chaotic map, the chaotic ability of encryption systems was improved in reference [18] and reference [27]. The result shows that by integrating different chaotic maps, the improved system has better sensitivity and larger key space.

As high-dimension chaotic systems were introduced to the encryption system, people mainly focused on the good encryption effects of high-dimension selection. But the initiated value decision which is also important was ignored. This paper presents the common flows and the sensitive flows of the maps used in image encryption. Then, it describes how to choose the sensitive flow of the chaotic system. The sensitive flow is used to generate random sequences in this paper. Experiment results and analysis prove that the image disposed by this algorithm has ability to resist brute-force attack, differential attack, statistic attack, chosen-plaintext attack, noise attack, and so on.

The rest of the paper is organized as follows: Section 2 introduces the preliminaries of the high-dimension system; Section 3 describes the encryption scheme with sensitive flows of the high-dimension system; Section 4 presents the experimental performance evaluation and analysis; Section 5 is the summary part.

2. Preliminaries

High-dimension systems have complex performance in space. To describe the character well, the Lorenz system is used as an example to do the analysis.

2.1. Three-Dimension Lorenz System. The Lorenz system is defined as

$$F(x_{n+1}, y_{n+1}, z_{n+1}) = \begin{cases} \delta(y_n - x_n), \\ r x_n - y_n - x_n z_n, \\ x_n y_n - \varepsilon z_n. \end{cases} \quad (1)$$

The Lyapunov exponents (LE) are shown in Figure 1. Figure 1(a) presents the LE value of δ in the interval [0, 100], while keeping $r = 8/3$ and $\varepsilon = 26$. Figure 1(b) refers to the LE value of r in the interval [0, 100], while keeping $\delta = 10$ and $\varepsilon = 26$. Figure 1(c) shows the LE value of ε in the interval [0, 100], while keeping $\delta = 10$ and $r = 8/3$. The Lorenz system has good chaotic behavior when $\delta = 10, r = 8/3$, and $\varepsilon = 26$ because the LE values of the three parameters are all positive at that point. Figure 1 shows that the chaotic behavior happens not just at that point. As long as the parameters' LE satisfy LE1>0, LE2>0, and LE3>0, where LE1 denotesδ's LE, LE2 means r's LE, and LE3 stands for ε's LE.

2.2. Manifold Computing for Sensitive Flows. Chaotic maps have good ability to generate a sequence with little correlation. But there is no reference describing the essence of the sequence used for encryption. In fact, the sequences are discrete points on flows in manifold of the maps. Taking Lorenz equations as an example, we find no analytical solutions of Lorenz equations in three-dimension space. The only way to describe the system is by numerical fitting by equation (1). Figure 2(a) shows the global manifold of the Lorenz system in three-dimension space by numerical fitting. The global manifold consists of lots of flows starting from the local manifold, as show in Figure 2(b). The sequences that we used for encryption are just a segment of one flow after hundred times iteration by equation (1) with an initiate value. Equation (2) defines the manifold of the chaotic system:

$$W^s(S_0) := \left\{ x \in R^n \,\middle|\, \lim_{t \longrightarrow \infty} \varphi^t(x) = S_0 \right\} = \lim_{t \longrightarrow -\infty} \varphi^t\left(W^s_{\mathrm{loc}}(S_0)\right), \quad (2)$$

where $W^s(S_0)$ is the manifold, $\phi(x)$ is the equation of the system, $W^s_{\mathrm{loc}}(S_0)$ is the local manifold, S_0 is the singular point of the system, and $t \longrightarrow \infty$ stands for infinite iterations.

From Figure 1(c) and Table 1, we can see that the flows starting from the local manifold are not in uniform distribution. Most flows are from a little angle range as shown in reference [5, 28, 29]. This means that the flows starting from a small angle range have less correlation compared with others. It also shows that the flows from the small angle range are sensitive. It has been proved in reference [29] that the sensitive flows start from the direction of the eigenvector of the manifold. So, the small angle range is just the directions of eigenvectors. The eigenvalues and eigenvectors of the Lorenz system are given in the following equation:

$$\lambda = \begin{pmatrix} -22.8277 & 0 & 0 \\ 0 & 11.8277 & 0 \\ 0 & 0 & -2.6667 \end{pmatrix}, \quad V = \begin{pmatrix} -0.6148 & -0.4165 & 0 \\ 0.7887 & -0.9091 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3)$$

The local manifold can be decided as

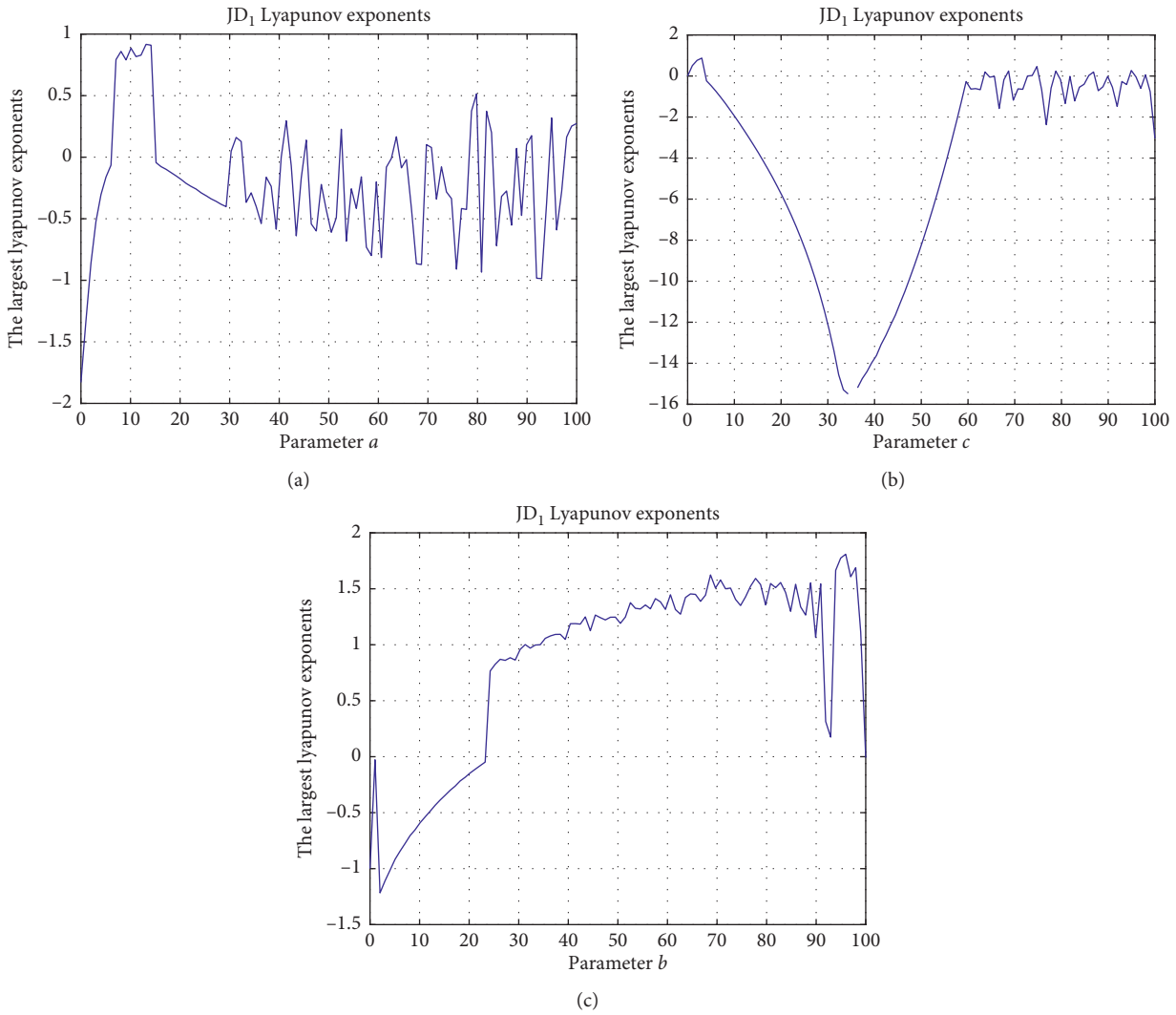

Figure 1: The LE values of the Lorenz system.

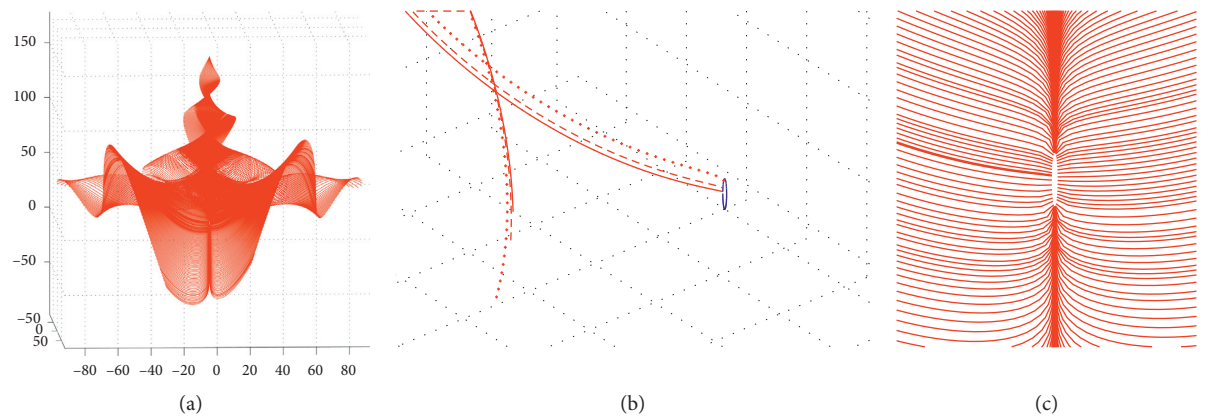

Figure 2: The global manifold and the flows.

Table 1: The flows with the changing angle.

Angle	Angle changing sensitivity (radian)	Flow number
0.0000–1.5708	10^{-1}–10^{-6}	41
1.5708–1.5709	10^{-6}–10^{-16}	578
1.5908–4.7124	10^{-1}–10^{-6}	35
4.7124–4.7125	10^{-6}–10^{-16}	193
4.7125–6.2842	10^{-1}–10^{-6}	25

$$x(S_0) = S_0 + \delta\left(\cos(\theta)\frac{\vec{V}_1}{\lambda_1} + \sin(\theta)\frac{\vec{V}_3}{\lambda_3}\right). \quad (4)$$

where $\vec{V}_1$ and $\vec{V}_3$ are the eigenvectors of the manifold and λ_1 and λ_3 are the corresponding eigenvalues.

The flows starting from the local manifold can be computed as

$$\Delta\theta = \begin{cases} \Delta\theta * 1.01, & d < 0.01 * \Delta d, \\ \Delta\theta + 2 * \Delta\theta * \left(\dfrac{d}{\Delta d}\right), & 0.01 * \Delta d \le d < 0.5 * \Delta d, \\ \Delta\theta, & 0.5 * \Delta d \le d < \Delta d, \\ \Delta\theta - \Delta\theta * \left(\dfrac{d}{(10 * \Delta d)}\right), & \Delta d \le d < 10 * \Delta d, \\ \dfrac{\Delta\theta}{\lg(d/\Delta d)}, & 10 * \Delta d \le d. \end{cases} \quad (5)$$

For every encryption chaotic system, scrambling and diffusing by the sequences of the sensitive flows can achieve better effect. The way to get the most sensitive flows follows the steps given below (take the Lorenz system as an example):

Step 1. Get the eigenvalues and eigenvectors as equation (3) by the Jacobian matrix of the system.

Step 2. Decide the local manifold by equation (4), as seen as the blue ellipse in Figure 2(b).

Step 3. Choose the initiate value on the ellipse in the eigenvector direction and do iteration along that flow. The initiate value sensitivity is 10^{-16} for the Lorenz system.

Step 4. Choose a segment of the flow computed in step 3 and have the right points which the key sequence need.

The sequence of the chaotic map from the sensitive flows of the maps for the encryption system can be obtained.

As can be seen from Figure 2, the global manifold is composed of flows in two small angle regions (1.5708–1.5709 and 4.7124–4.7125). The flows of these two regions are intertwined in space but never intersect. After these streams are discretized, these discrete points are of confusion and complexity, which are convenient to become the initial sequence of the encryption system.

3. The Encryption and Decryption Algorithms

Image encryption refers to pixel transformation based on different schemes. The transformation can be divided into position scrambling and value diffusion, generally both position and value of pixels will be changed in the encryption process. One scheme of the transformation indicates finding the right ways to choose sequences, which will be used to position scrambling and value diffusion. The less the relationship among the sequence, the harder for the attackers to find rules of the encryption. The chaotic system can generate sequence with little relationship. Another scheme for transformation is to get the right computation algorithm, which can do computation with pixels and sequences. In the paper, the high-dimensional manifold computing will generate a sequence and block dividing algorithm which can do transform computations.

The encryption process is as shown in Figure 3. It contains sensitive flow computing to produce key sequences, two scrambling operations, and one divided blocks matching diffusion operation.

3.1. The Encryption Algorithm. The encryption process follows the steps given below:

Step 1. Generating sequences: by selecting sensitive flows of the manifold, we can get three sequences S_1, S_2, and S_3 with the size $(M+m) \times (N+n)$ where M is the number of rows of the image and N refers to the column numbers of the image. $M+m$ satisfies $(M+m)\text{mod}4 = 0$ and $N+n$ satisfies $(N+n)\text{mod}\,4 = 0$. Transformation is performed using the following equations:

$$\begin{aligned} S_1' &= \left(\text{round}\left(S_1 * 10^{15}\right)\right)\text{mod}\,255, \\ S_2' &= \left(\text{round}\left(S_2 * 10^{15}\right)\right)\text{mod}\,255, \\ S_3' &= \left(\text{round}\left(S_3 * 10^{15}\right)\right)\text{mod}\,255. \end{aligned} \quad (6)$$

Step 2. Image scrambling: the scrambling matrices will be produced by the following equation:

$$\begin{aligned} W_l(i,j) &= \begin{cases} 1, & \text{for}\,(S_2'(i * (M+m) + j), j), \\ 0, & \text{others}, \end{cases} \\ Wr(i,j) &= \begin{cases} 1, & \text{for}\,(i, S_2'(j * (N+n) + i)), \\ 0, & \text{others}. \end{cases} \end{aligned} \quad (7)$$

The image I was transformed to I' by expanding the size as $(M+m) \times (N+n)$. Then, the scrambling can be finished by

$$I'' = W_l^T I' W_r^T. \quad (8)$$

Step 3. Block matching: we divide image I'' into 16 blocks $(I_1'', I_2'', \ldots, I_{16}'')$. The sequence S_2' was changed into matrix S_2'' and then divided into 16 blocks $S_2'' = (S_2''(1), S_2''(2), \ldots, S_2''(16))$. For each block, it has

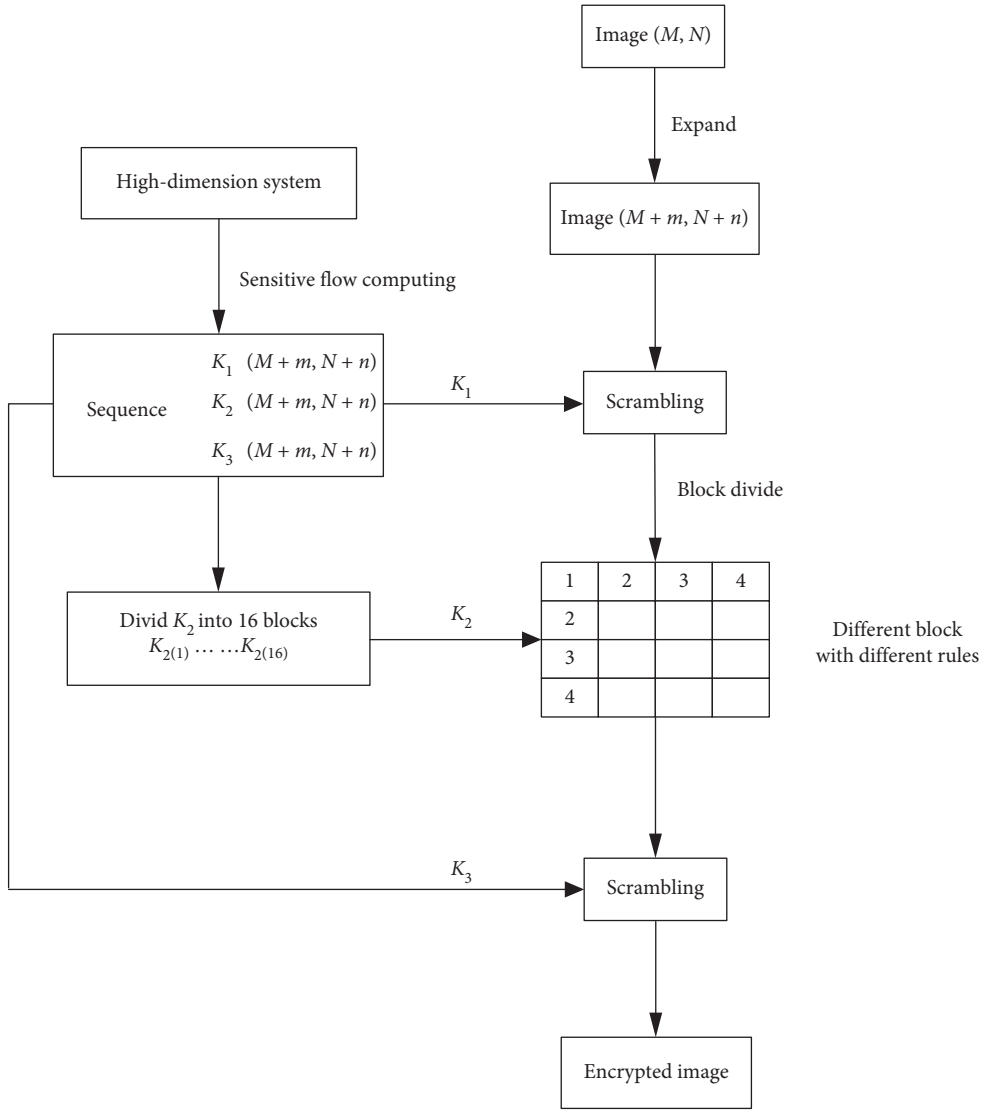

Figure 3: The encryption process.

the size of $(M+m)/4$, $(N+n)/4$. We also have different diffusion rules for each block as the following:

$$I_i'' \oplus \left(\left(i * S_2''(i) \right) \right) \bmod 255. \tag{9}$$

The pixels in different blocks can be encrypted by different rules. After the diffusion section, we can get the new image $I_{\text{diffusion}}$.

Step 4. Image scrambling: the scrambling matrices will be produced by the following equation:

$$W_l'(i,j) = \begin{cases} 1, & \text{for} \left(S_3'(i * (M+m) + j), j \right), \\ 0, & \text{others}, \end{cases}$$

$$W_r'(i,j) = \begin{cases} 1, & \text{for} \left(i, S_3'(j * (N+n) + i) \right), \\ 0, & \text{others}, \end{cases} \tag{10}$$

$$I_{\text{encrypted}} = W_l'^{T} I_{\text{diffusion}} W_r'^{T}.$$

3.2. The Decryption Process. Decryption is the reverse of encryption. It can be finished as given in the following steps:

Step 1. Encrypted image scrambling: by equation (11), we can get the diffusion image:

$$I_{\text{diffusion}} = \left(W_l'^{T} \right)^{-1} I_{\text{encrypted}} \left(W_r'^{T} \right)^{-1}. \tag{11}$$

Step 2. We divide the diffusion image into 16 blocks $I_{\text{diffusion}}(1), I_{\text{diffusion}}(2), \ldots, I_{\text{diffusion}}(16)$ and do block decryption with different rules as following:

$$I_{\text{diffusion}}(i) \oplus \left(i * S_2''(i) \right) \bmod 255. \tag{12}$$

After the block decryption by equation (12), we can get image I''.

Step 3. Image scrambling: by equation (13), we can restore the expanded image I':

$$I' = \left(W_l^T\right)^{-1} I'' \left(W_r^T\right)^{-1}. \quad (13)$$

Step 4. By removing the redundant rows and columns, we will get the plain image I in normal size.

4. Experiment and Analysis

The encryption algorithm should withstand different kinds of attacks such as brute-force attack, statistical attack, differential attack, chosen-plaintext attack, chosen chipper text attack, known plaintext attack, and known chipper attack. Different kinds of attack refer to different kinds of performance indicators of the encryption system. To be easily compared to the results in references, "lena," "baboon," "peppers," and "airplane" are chosen as the plain text in this paper. They all have the same size $512 * 512$. Several experiments and different kinds of security analysis are presented to evaluate the robustness of the proposed encryption method.

4.1. Statistical Analysis. Histogram is always used for statistical analysis. A histogram is a graphical method for displaying the image's exposure accuracy by using the graphical parameter and describes image's distribution curve. As we all know that the confusion operation does not change the histogram, but the diffusion makes the distribution of the histogram uniform. Taking the image "airplane" as an example, we can see the results as following.

Figure 4(a) is the original image, and Figure 4(d) is the histogram of the original image. Figure 4(b) is the confusion image, and Figure 4(e) is the histogram of the confusion image. We can see that the scrambling operation did not change the histogram distribution. Figure 4(c) is the diffusion image, and Figure 4(f) is the histogram of the diffusion image. The distribution of the histogram is uniform in Figure 4(f), which means the pixels value is changed to be equal in numbers.

4.2. Correlation Analysis. A good encryption algorithm should have the ability to break correlations between adjacent pixels. It may cause information leakage by statistical attacks if this feature is ignored. The correlation coefficient of an image is calculated as

$$\begin{aligned} E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)^2, \\ \operatorname{cov}(x,y) &= \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)\left(y_i - E(y)\right), \\ \gamma &= \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)} \times \sqrt{D(y)}}, \end{aligned} \quad (14)$$

where x, y denote the pixel values of two adjacent pixels, $E(x)$ is the function for mean value, $D(x)$ is for the variance and $\operatorname{cov}(x, y)$ is the function for correlation, and γ denotes the correlation coefficient of the image.

Taking image "lena" as an example, Figure 5(a) shows the correlation in the horizontal direction of the plain image. Figure 5(b) shows the correlation in the vertical direction of the plain image. Figure 5(c) shows the correlation in the diagonal direction of the plain image. We can see that in all the three directions, the pixels have strong correlation in the plain image. Figures 5(d)–5(f) show correlation in the horizontal direction, vertical direction, and diagonal direction of the cipher image. Pixels of the encrypted image in all three directions have little correlation, which means there are no rules for attackers to find.

Table 2 shows the correlation value in three directions of different images. It has proved that the novel encryption scheme proposed in this paper has generality for all images.

4.3. Key Sensitivity Analysis. Key sensitivity is an important feature for any good encryption system.

"The good sensitivity means that the encrypted image cannot get the right image even though there is a tiny change in the secret key." Taking the image "couple" as an example, tiny changes $\Delta = 10^{-15}$, $\Delta = 10^{-16}$, and $\Delta = 10^{-18}$ are chosen as the key disturbance, and the experiment results are shown as the following.

Figure 6(a) is the plain image, and Figure 6(b) is the cipher image. Figure 6(c) is the decrypted image with the right key, and Figure 6(d) is the decrypted image with tiny change $\Delta = 10^{-15}$. Figures 6(e) and 6(f) are the decrypted images with tiny changes $\Delta = 10^{-16}$ and $\Delta = 10^{-18}$. We can see that the slight change $\Delta = 10^{-15}$ leads to a complete wrong result, but $\Delta = 10^{-16}$ and $\Delta = 10^{-18}$ make the right result. It can be proved that the sensitivity of the encryption system is $\Delta = 10^{-15}$.

Table 3 shows the difference rate of the slightly changing key with different images. For all the images chosen as plain images, the sensitivity always keeps as $\Delta = 10^{-15}$.

4.4. Key Space Analysis. The key space reflects the ability of the encryption algorithm to resist brute-force attack. If the key space is larger than 2^{128}, it means the system can stand for brute-force attack. It has been proved that the sensitivity is $\Delta x = 10^{-15}$. The sensitive flow computing make the space to be 10^{88}. The encrypted system is three dimensional with two confusion steps. The image is divided into 16 blocks in diffusion operation. So the key space can be calculated as

$$\text{keyspace} = 10^{88} * 2^3 * \left(2^4 * 2^3\right) * 2^3 \approx 2^{216}. \quad (15)$$

For $2^{216} \gg 2^{128}$, the presented schemes can resist brute-force attack.

4.5. Differential Attack Analysis. The following equations are used to analyze the effectiveness of the differential attack resistant:

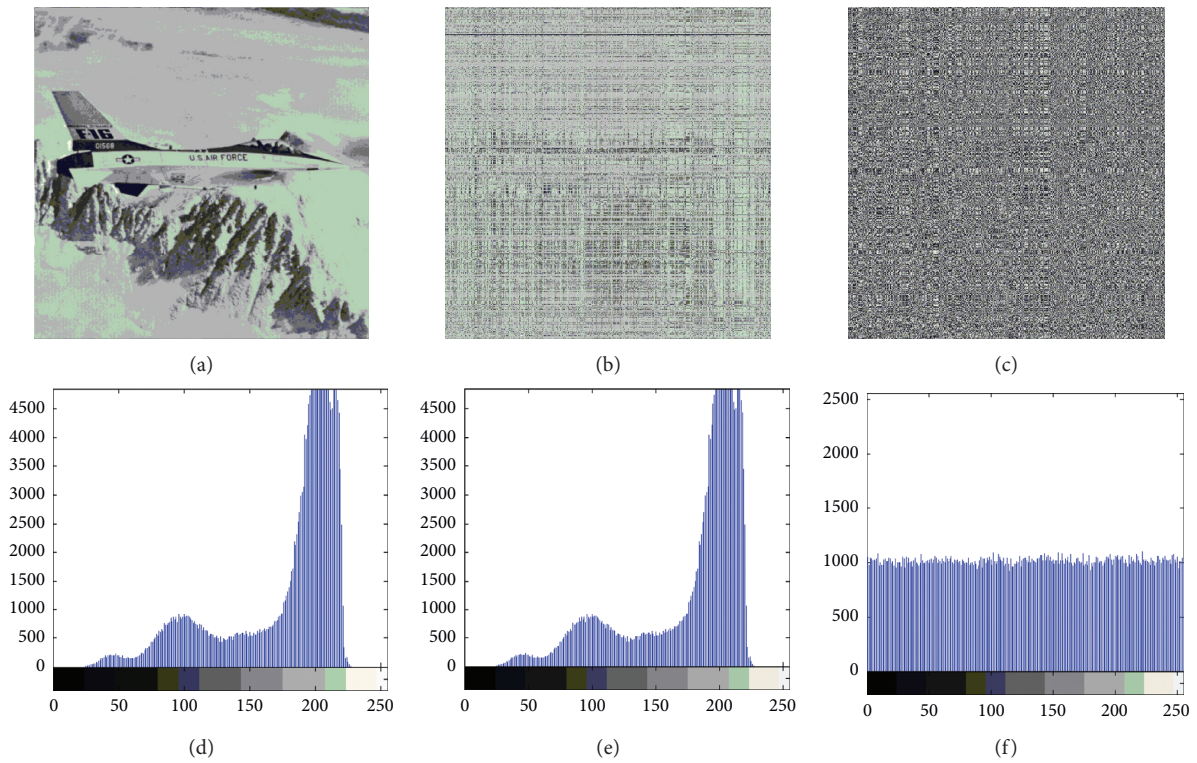

Figure 4: The histogram of the image "airplane."

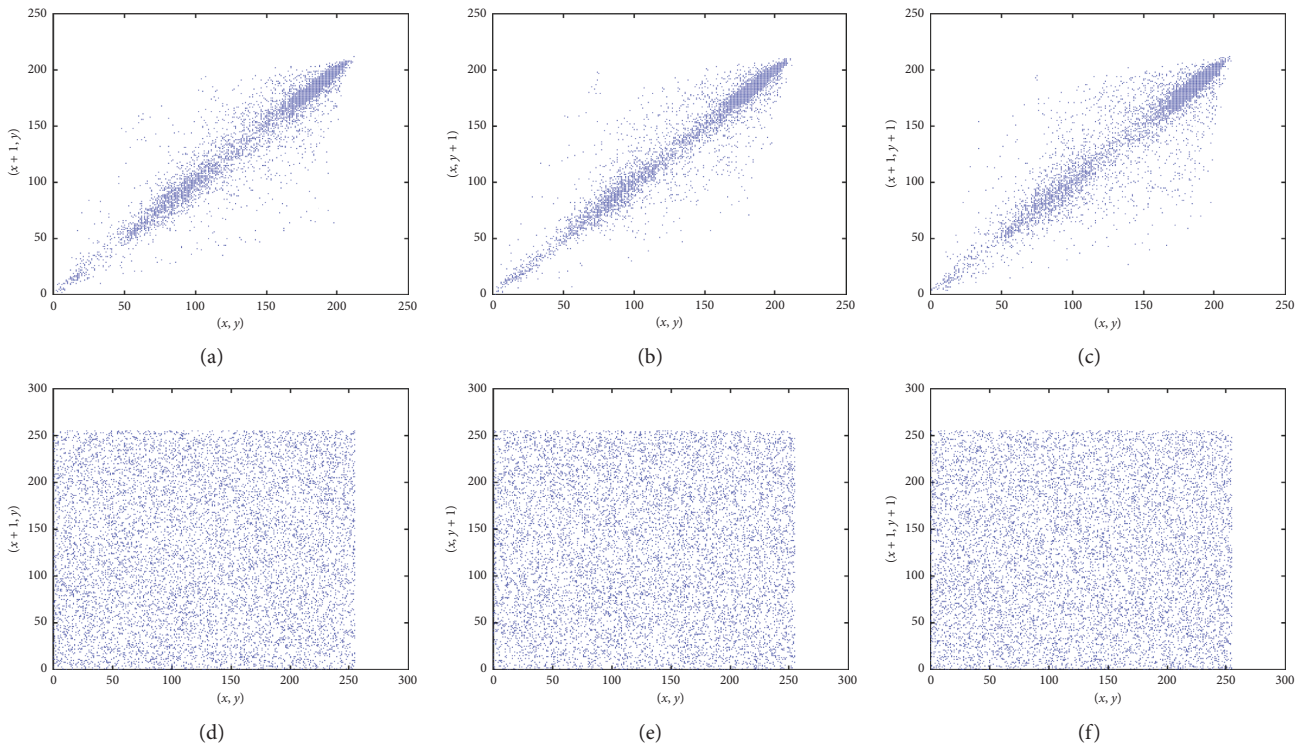

Figure 5: The correlation between X-X, Y-Y, and X-Y directions of the plain image and the cipher image.

Table 2: Correlation coefficients between the original and cipher images.

Plain/cipher image	Image	Correlation direction		
		Horizontal	Vertical	Diagonal
Plain image	Lena	0.9849	0.9718	0.9570
	Baboon	0.7574	0.8633	0.7293
	Airplane	0.9648	0.9465	0.9722
	Couple	0.9524	0.9155	0.9450
Cipher image	Lena	0.0068	0.0103	0.0044
	Baboon	0.0148	0.0248	0.0061
	Airplane	0.0069	0.0086	0.0133
	Couple	0.0380	0.0041	0.0205

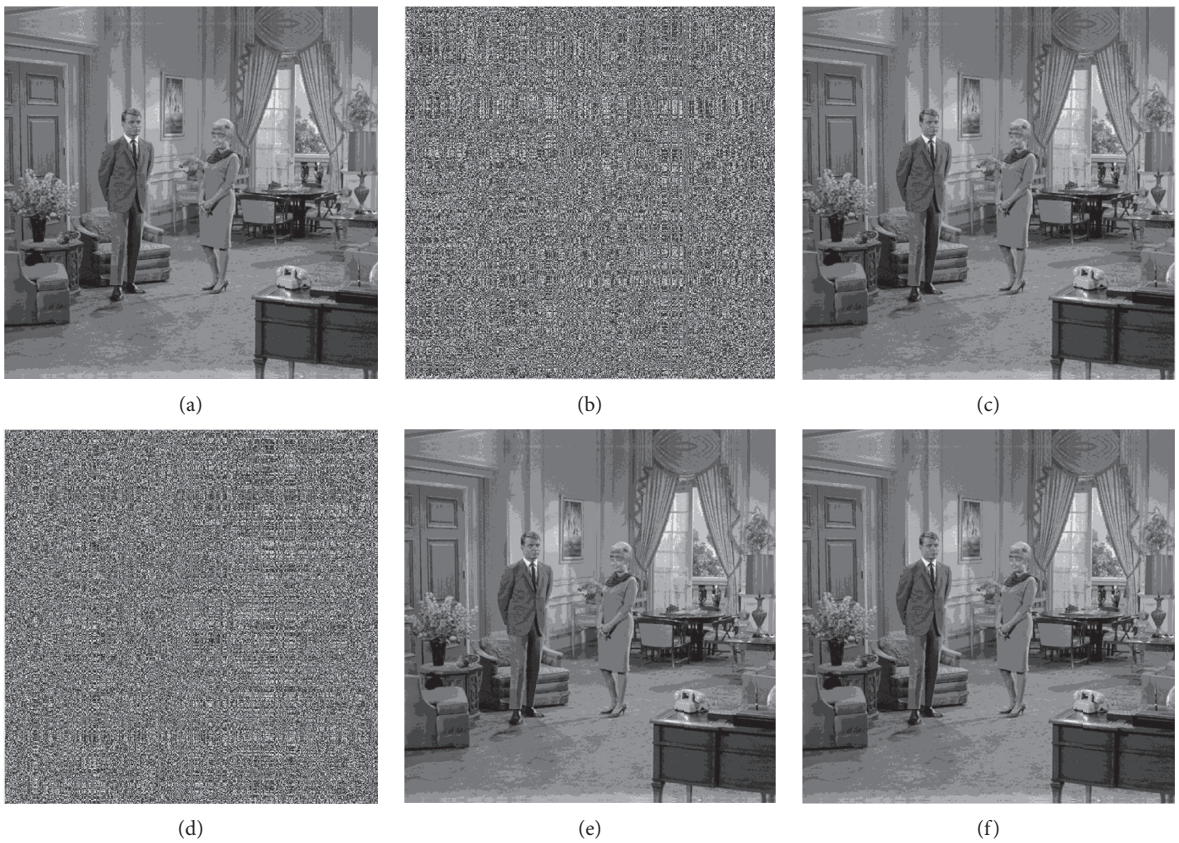

Figure 6: The plain image cipher image with the right key and the wrong key.

Table 3: The difference rate of slightly changing key with different images.

The changed key Δ	Difference rate			
	Couple	Lena	Airplane	Baboon
$K + \Delta = 10^{-15}$	99.585	99.591	99.615	99.599
$K + \Delta = 10^{-16}$	0	0	0	0
$K + \Delta = 10^{-18}$	0	0	0	0

Table 4: Comparison of NPCR and UACI with the image "lena."

Lena image	NPCR	UACI
By this scheme	99.61	33.43
Reference [2]	99.60	33.48
Reference [30]	99.6836	33.503
Reference [5]	99.6017	28.1370

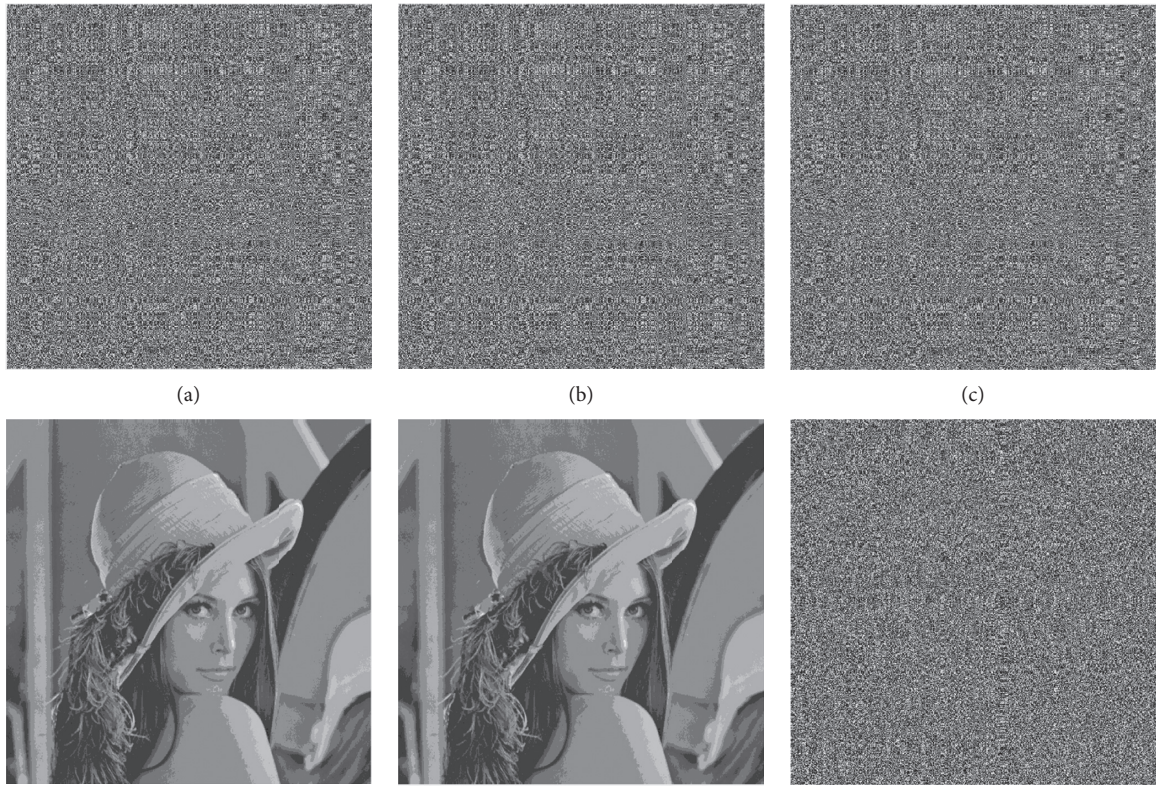

Figure 7: Noise attack and noised decryption.

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

$$D(i,j) = \begin{cases} 1 & \text{if } C \neq C' \\ 0 & \text{if } C = C' \end{cases}, \tag{16}$$

$$\text{UACI} = \frac{1}{M \times N} \left[\sum_{i,j} \frac{|I - I'|}{255} \right] \times 100\%,$$

where NPCR refers to the number of pixels changed, UACI stands for the average changing intensity, $D(i,j)$ is the Boolean function, M is equal to the row number of each color field matrix, N is equal to the line number of each color field matrix, and I is the encrypted plain text, while I' is one pixel changed encrypted plain text.

Table 4 shows the comparison of NPCR and UACI of the image "lena" with the method in this paper and the methods in the references. It demonstrates that our encryption method has strong ability of withstanding the differential attack. We got a larger NPCR value than that in Reference [30] and Reference [5] and a slightly smaller UACI value than that of Reference [30] and Reference [5].

4.6. Noise Attack. An efficient image encryption algorithm should be robust against different kinds of the noise. In this experiment, we first add salt and pepper noise, speckle noise, and Gaussian noise with density 0.000001 to the cipher image to see if the algorithm can decrypt the right plain text.

Figure 7(a) shows the salt and pepper noise which was added to the encrypted image, and Figure 7(d) shows the decryption result. Figure 7(b) shows the speckle noise which was added to the encrypted image, and Figure 7(e) shows the decryption result. Figure 7(c) shows the Gaussian noise which was added to the encrypted image, and Figure 7(f) shows the decryption result. We can see that the scheme can fight agaist salt and pepper noise and speckle noise and can resist the Gaussian noise attack.

Table 5: The Shannon entropy analysis of different images.

Image	Lena	Baboon	Couple	Airplane
Plain image	7.2185	7.1391	7.0517	6.7178
Cipher image	7.9985	7.9875	7.9882	7.997

Table 6: Quality evaluation metrics of decrypted images.

Image	PSNR	NCC	SSIM
Lena	∞	1	1
Baboon	∞	1	1
Couple	∞	1	1
Airplane	∞	1	1

4.7. Information Entropy Analysis. Information entropy is employed to measure the uncertainty in a random variable, which can be described as

$$H(I) = -\sum_{i=0}^{255} P_r(I = i)\log p_{r(I=i)}, \tag{17}$$

where $p_r(I = i)$ represents the probability of the symbol i. For an 8 bit truly random image, the ideal entropy is 8. The closer $H(I)$ gets to 8, the better the randomness of I. Table 5 shows the Shannon entropy analysis of different images by this algorithm.

4.8. Quality Evaluation Metrics of Decryped Image. A general requirement for all image encryption schemes is that the encrypted image should be greatly different from the plain image, while the decrypted image should be as same as the plain image. We use peak signal-to-noise ratio (PSNR) to measure the difference between the plain image and the decrypted image and the normalized cross correlation (NCC) and structural similarity index metric (SSIM) to see if there is information leakage in the encryption process:

$$\begin{aligned}
\mathrm{PSNR} &= 10\log_{10}\frac{D^2\max}{\mathrm{MSE}},\\
\mathrm{MSE} &= \frac{1}{M * N}\sum_{x=1}^{M}\sum_{y=1}^{N}(I(x,y) - D(x,y)),\\
\mathrm{NCC} &= \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(I(x,y) * D(x,y))}{\sum_{x=1}^{M}\sum_{y=1}^{N}(I(x,y))^2},\\
\mathrm{SSIM}(I,D) &= \frac{(2\mu_1\mu_2 + C_1)(2\delta_{12} + C_2)}{(\mu_1^2 + \mu_2^2 + C_1)(\delta_1^2 + \delta_2^2 + C_2)}.
\end{aligned} \tag{18}$$

The PSNR is used to see if the encrypted image contains useful information. The higher the peak signal-to-noise ratio, the more confusing the encrypted image. If the peak signal-to-noise ratio approaches infinity, the encrypted image approximates to noise. NCC and SSIM are used to measure whether the decrypted encrypted image was the same as the original plain text image. Their intervals are [0, 1], and if their values are 1, then the information is 100% reserved in the both encryption and decryption process. This indicator measures the efficiency and accuracy of the decryption process [31–33].

$I(x, y)$ is the plain image. $D(x, y)$ refers to the decrypted image. M and N stand for the row numbers and the column numbers. μ_1 and δ_1 are the mean value and variance value of $I(x, y)$, while μ_2 and δ_2 are the mean value and variance value of $D(x, y)$. δ_{12} is the covariance of $I(x, y)$ and $D(x, y)$. C_1 and C_2 are constants to avoid zero value of the denominator. We made $C_1 = 2.25$ and $C_2 = 2.25 * 3$ in this experiment.

From Table 6, we can see that PSNR is close to ∞, which means the plain image and the decrypted image are completely the same. NCC and SSIM are both equal to 1, which means there is no information loss through the encryption and decryption process.

5. Conclusion

In this paper, the sensitive flows of the manifold in the chaotic system which is always ignored were presented first. The way to get the sensitive flows was shown then. By the sensitive flows, the sequences were generated as the keys of the encryption system. The encryption process contains two scrambling steps and one diffusion step. In the diffusion section, the image was divided into 16 blocks, and different encrypted rules were used for different blocks. In this way, the key space was improved and the correlation was reduced. In the encryption process, the high-dimensional manifold is taken to do calculation, and only the local manifold region with the best sensitivity is used as the initial secret key to generate the pseudorandom sequence. The scrambled image is processed in blocks, and different random sequences are used to encrypt different regions, which makes the encryption system more complicated. The decryption process is the reverse of the encryption process. The final original image can only be obtained by partitioning the ciphertext image and then restoring the position. Simulation results and analysis show that the novel algorithm has ability to encrypt the image into random-like cipher images. It can also resist common kinds of attack.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This research was jointly supported by the National Natural Science Foundation of China (Grant 61501391) and Natural Science Foundation of Henan Province (182300410258).

References

[1] J. Wu, X. Liao, and B. Yang, "Color image encryption based on chaotic systems and elliptic curve ElGamal scheme," *Signal Processing*, vol. 141, pp. 109–124, 2017.

[2] S. Toughi, M. H. Fathi, and Y. A. Sekhavat, "An image encryption scheme based on elliptic curve pseudo random and advanced encryption system," *Signal Processing*, vol. 141, pp. 217–227, 2017.

[3] P. Ping, J. Wu, Y. Xu et al., "Design of image cipher using life-like cellular automata and chaotic map," *Signal Processing*, vol. 150, pp. 233–247, 2018.

[4] X. Q. Fu, B. C. Liu, and Y. Y. Xie, "Image encryption-then-transmission using DNA encryption algorithm and the double chaos," *IEEE Photonics Journal*, vol. 10, Article ID 3900515, 2018.

[5] H. Liu, X. Wang, and A. kadir, "Image encryption using DNA complementary rule and chaotic maps," *Applied Soft Computing*, vol. 12, no. 5, pp. 1457–1466, 2012.

[6] X. Wei, L. Guo, Q. Zhang, J. Zhang, and S. Lian, "A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system," *Journal of Systems and Software*, vol. 85, no. 2, pp. 290–299, 2012.

[7] R. Enayatifar, A. H. Abdullah, and I. F. Isnin, "Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence," *Optics and Lasers in Engineering*, vol. 56, pp. 83–93, 2014.

[8] X. Wu, K. Wang, X. Wang et al., "Color image DNA encryption using NCA map-based CML and one-time keys," *Signal Processing*, vol. 148, pp. 272–287, 2018.

[9] X. Chai, X. Zheng, Z. Gan et al., "An image encryption algorithm based on chaotic system and compressive sensing," *Signal Processing*, vol. 148, pp. 124–144, 2018.

[10] Y. Zhang, Y. Li, W. Wen, Y. Wu, and J. Chen, "Decipheringan image cipher based on 3-cell chaotic map and biological operations," *Nonlinear Dynamics*, vol. 82, pp. 1831–1837, 2016.

[11] Y. Zhang, "The unified image encryption algorithm based on chaos and cubic S-Box," *Information Sciences*, vol. 450, pp. 361–377, 2018.

[12] W.-K. Lee, R. C.-W. Phan, W.-S. Yap, and B.-M. Goi, "SPRING: a novel parallel chaos-based image encryption scheme," *Nonlinear Dynamics*, vol. 92, no. 2, pp. 575–593, 2018.

[13] Y. Luo, R. Zhou, J. Liu, Y. Cao, and X. Ding, "A parallel image encryption algorithm based on the piecewise linear chaotic map and hyper-chaotic map," *Nonlinear Dynamics*, vol. 93, no. 3, pp. 1165–1181, 2018.

[14] A. Yannick and T. Alain, "Image encryption by chaos mixing," *Iet Image Processing*, vol. 10, pp. 742–750, 2016.

[15] Z. Hua and Y. Zhou, "Design of image cipher using block-based scrambling and image filtering," *Information Sciences*, vol. 396, pp. 97–113, 2017.

[16] L. Chen, B. Ma, X. Zhao, and S. Wang, "Differential cryptanalysis of a novel image encryption algorithm based on chaos and Line map," *Nonlinear Dynamics*, vol. 87, no. 3, pp. 1797–1807, 2017.

[17] P. Ping, F. Xu, Y. Mao, and Z. Wang, "Designing permutation-substitution image encryption networks with Henon map," *Neurocomputing*, vol. 283, pp. 53–63, 2017.

[18] C. Cao, K. Sun, and W. Liu, "A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map," *Signal Processing*, vol. 143, pp. 122–133, 2018.

[19] Y.-G. Yang, J. Tian, H. Lei, Y.-H. Zhou, and W.-M. Shi, "Novel quantum image encryption using one-dimensional quantum cellular automata," *Information Sciences*, vol. 345, pp. 257–270, 2016.

[20] A. Souyah and K. Faraoun, "An image encryption scheme combining chaos-memory cellular automata and weighted histogram," *Nonlinear Dynamics*, vol. 86, pp. 1–15, 2016.

[21] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, "Image compression-encryption scheme based on hyper-chaotic system and 2D compressive sensing," *Optics & Laser Technology*, vol. 82, pp. 121–133, 2016.

[22] X. Chai, Z. Gan, Y. Chen, and Y. Zhang, "A visually secure image encryption scheme based on compressive sensing," *Signal Processing*, vol. 134, pp. 35–51, 2016.

[23] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.

[24] L. Teng and X. Wang, "A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive," *Optics Communications*, vol. 285, no. 20, pp. 4048–4054, 2012.

[25] C. Zhu, "A novel image encryption scheme based on improved hyperchaotic sequences," *Optics Communications*, vol. 285, no. 1, pp. 29–37, 2012.

[26] C. Zhu and K. Sun, "Cryptanalysis and improvement of a class of hyperchaos based image encryption algorithms," *Acta Physica Sinica*, vol. 61, no. 12, p. 120503, 2012.

[27] R. Lan, J. He, and S. Wang, "Integrated chaotic systems for image encryption," *Signal Processing*, vol. 147, pp. 133–145, 2018.

[28] M. Jia, Y. Y. Fan, and H. M. Li, "The computation of invariant manifolds with self-adaptive parameter and trajectories continuation method," *Acta Physica Sinica*, vol. 59, pp. 7686–7692, 2010.

[29] M. Jia, Y. Y. Fan, and W. J. Tian, "A fast computing method to distinguish the hyperbolic trajectory of an non-autonomous system," *Chinese Physics B*, vol. 20, Article ID 034701, 2011.

[30] M. Mollaeefar, A. Sharif, and M. Nazari, "A novel encryption scheme for colored image based on high level chaotic maps," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 607–629, 2017.

[31] A. Belazi, A. A. Abd El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Processing*, vol. 128, pp. 155–170, 2016.

[32] N. A. Mohamed, M. A. E. Azeim, and A. Zaghloul, "Image encryption scheme for secure digital images based on 3D cat map and turing machine," in *Proceedings of the 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, Fukuoka, Japan, November 2015.

[33] A. Belazi and R. Rhouma, "Selective image encryption scheme based on DWT, AES S-box and chaotic permutation," in *Proceedings of the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Dubrovnik, Croatia, August 2015.