Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 9524080, 12 pages
https://doi.org/10.1155/2019/9524080

Research Article

Image Encryption Algorithm Based on the H-Fractal and Dynamic Self-Invertible Matrix

Xuncai Zhang, Lingfei Wang, Ying Niu, Guangzhao Cui, and Shengtao Geng

School of Electrics and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Correspondence should be addressed to Ying Niu; niuying@zzuli.edu.cn

Received 2 March 2019; Revised 27 April 2019; Accepted 20 May 2019; Published 13 June 2019

Academic Editor: Amparo Alonso-Betanzos

In this paper, an image encryption algorithm based on the H-fractal and dynamic self-invertible matrix is proposed. The H-fractal diffusion encryption method is firstly used in this encryption algorithm. This method crosses the pixels at both ends of the H-fractal, and it can enrich the means of pixel diffusion. The encryption algorithm we propose uses the Lorenz hyperchaotic system to generate pseudorandom sequences for pixel location scrambling and self-invertible matrix construction to scramble and diffuse images. To link the cipher image with the original image, the initial values of the Lorenz hyperchaotic system are determined using the original image, and it can enhance the security of the encryption algorithm. The security analysis shows that this algorithm is easy to implement. It has a large key space and strong key sensitivity and can effectively resist plaintext attacks.

1. Introduction

In modern society, technologies such as the Internet and block-chains are rapidly developing, and human beings have entered the big data era. Internet technology has brought great convenience to human life and promoted the establishment of global information access. With the development of multimedia technology, digital offices and electronic payments have become more popular in various fields of human life. Compared with textual information, the informative features that are expressed by images are more intuitive, and the amount of information that images contain has increased. At this stage, images are being used as the main carrier of information. While enjoying the convenience brought by the information society, we must also be more vigilant about the disasters that can be caused by information leakage. For example, in June 2013, former CIA employee Snowden revealed the "PRISM Project" to the world. Some high-tech companies with great influence left back doors in the equipment that they produced, making it convenient for the US government to monitor the public. During the Korean Winter Olympics in January 2018, the identity information and bank account information of a large number of athletes and spectators were maliciously acquired by hackers, thereby causing adverse effects. Protecting the security of information and avoiding losses due to information leakage is an urgent task for human beings. Traditional encryption algorithms such as DES [1–3] and RSA [4–9] have a wide range of applications in text encryption, but the applications of traditional encryption algorithms are not sufficient to meet the timeliness and security requirements of image information encryption. Therefore, how to encrypt image information quickly and effectively has become a popular research field.

There are two main types of methods in image encryption algorithms: scrambling [10–15] and diffusion [16–20]. Scrambling is achieved by transforming the positions of the pixels. Transforming the positions of the pixels can decrease the correlation between adjacent pixels and achieve encryption. For example, in 2004, Maniccam et al. proposed an encryption method based on SCAN mode in which the image is encrypted by using a different scanning path [21]. In 2010, Jolfaei et al. proposed an encryption algorithm based on the Henon chaotic system that uses the sorting transformation method to encrypt images. This method makes cipher images more pseudorandom [22]. In 2011, Zhu proposed an encryption method based on bit-plane scrambling [10]. Diffusion is performed by changing

the values of the pixels. Diffusion encryption can enhance the randomness and break the statistical characteristics of the cipher images. For example, in 2009, Acharya proposed an encryption algorithm based on the Hill matrix that uses an invertible matrix to encrypt images [23]. In 2011, El-Zoghdy proposed an improved DES algorithm to encrypt images [24]. In recent years, some hybrid image encryption algorithms have been proposed. For example, in 2004, Gehani proposed an encryption method using DNA strings that applied DNA coding to image encryption [25]. In 2005, Guan proposed an encryption algorithm based on Arnold-Chen chaotic sequences that combined scrambling and diffusion in the image encryption process [26]. In 2008, Tong et al. proposed an encryption method that combined cyclic shifts and sequence encryption [27].

In this paper, an image encryption algorithm based on the H-fractal structure and dynamic self-invertible matrix is proposed. This algorithm combines the scrambling and diffusion encryption methods. Section 2 introduces the basic theory of this algorithm, Section 3 introduces the encryption scheme, and the security analyses of this encryption algorithm are given in Section 4. The results of the security analysis show that the encryption algorithm has good security, and it can be applied in the field of image encryption.

2. Fundamental Theory

2.1. Lorenz Hyperchaotic System. Chaotic systems are widely used in the information encryption field because their initial values and parameters are sensitive and pseudorandom [28, 29]. Low-dimensional chaotic systems have small key spaces and weak pseudorandomness. Therefore, many scholars have improved upon low-dimensional chaotic systems by developing chaotic systems to higher dimensions. These improved high-dimensional chaotic systems are called hyperchaotic systems. To generate the four pseudorandom sequences that are required by the encryption algorithm, we apply the Lorenz hyperchaotic system [30] to the encryption algorithm. The Lorenz hyperchaotic system is described as

$$\begin{cases} \dot{x} = a(y - x) + w, \\ \dot{y} = cx - y - xz, \\ \dot{z} = xy - bz, \\ \dot{w} = -yz + rw, \end{cases} \tag{1}$$

where a, b, c, and r are the four parameters of the Lorenz hyperchaotic system. When $a = 10$, $b = 8/3$, $c = 28$, and $-1.52 \le r \le 0.06$, the Lorenz hyperchaotic system is in a hyperchaotic state. The hyperchaotic system is iterated by using the Runge–Kutta method when $r = -1$. The simulation results of the Lorenz hyperchaotic system are shown in Figure 1.

2.2. Self-Invertible Matrix Encryption. In 1929, Hill proposed an encryption algorithm that used invertible matrices [31]. The fundamental theory of the algorithm is to use a matrix to convert the plain-text into cipher-text, and the key is the matrix itself. The encryption method is described as

$$C = KM \pmod{R}. \tag{2}$$

In formula (2), M represents a plain-text matrix, C represents a cipher-text matrix, R is the plain-text value range (in the image encryption process, $R = 256$), K represents an encryption key, and matrix K must be an invertible matrix. The Hill encryption algorithm is uncompressed. Assuming that the length of the plain-text and cipher-text is l, the encryption formula can also be described as

$$\begin{cases} c_1 = k_{11}m_1 + k_{12}m_2 + \ldots + k_{1l}m_l \pmod{R}, \\ c_2 = k_{21}m_1 + k_{22}m_2 + \ldots + k_{2l}m_l \pmod{R}, \\ \ldots\ldots \\ c_l = k_{l1}m_1 + k_{l2}m_2 + \ldots + k_{ll}m_l \pmod{R}. \end{cases} \tag{3}$$

In formula (3),

$$C = \begin{bmatrix} c_1 \\ c_2 \\ \ldots \\ c_l \end{bmatrix}, \quad M = \begin{bmatrix} m_1 \\ m_2 \\ \ldots \\ m_l \end{bmatrix}, \quad K = \left(k_{ij}\right)_{l \times l} = \begin{bmatrix} k_{11} & \ldots & k_{1l} \\ k_{21} & & k_{2l} \\ \ldots & \ldots & \ldots \\ k_{l1} & \ldots & k_{ll} \end{bmatrix}. \tag{4}$$

The decryption process is the inverse of formula (3) and can be described as

$$M = K^{-1}C \pmod{R}. \tag{5}$$

To ensure the existence of matrix K^{-1}, this paper constructs matrix K as a 4×4 self-reversible matrix so that $K^{-1}K \pmod{R} = E$. The decryption process can be simplified as

$$M = K^{-1}C \pmod{R} = KC \pmod{R}. \tag{6}$$

The method of calculating a 4×4 self-invertible matrix is as follows. When matrix A is a self-invertible matrix, $A^{-1}A \pmod{R} = E$. If $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, A_{11}, …, A_{22} are 2×2 matrices, and formula (7) can be derived:

$$AA^{-1} \pmod{R} = AA \pmod{R} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} * \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \pmod{R} = E. \tag{7}$$

Then, formula (8) can be calculated by expanding formula (7):

$$A_{12}A_{21} = E - A_{11}^2 = (E - A_{11})(E + A_{11}). \tag{8}$$

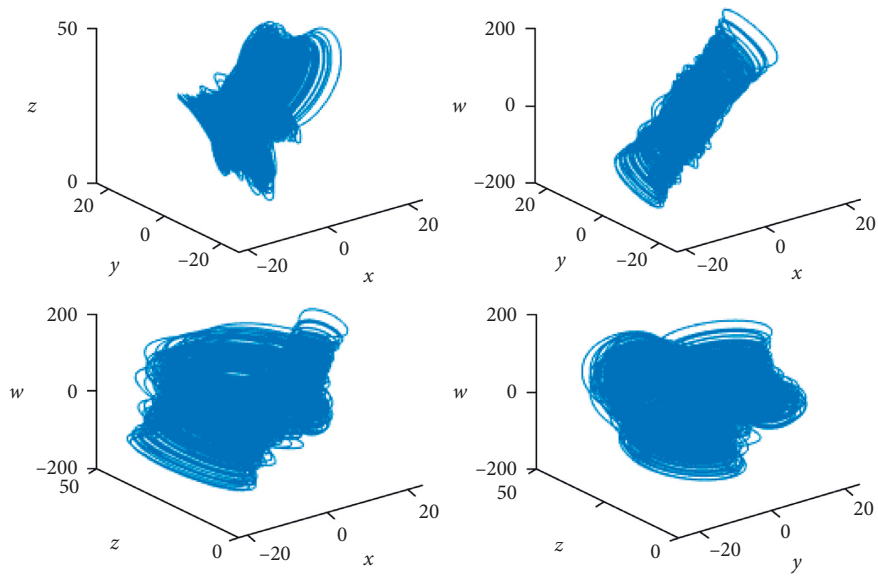

Figure 1: The phase diagram of the Lorenz hyperchaotic system.

To construct the self-invertible matrix, A_{12} is constructed as a factor of $(E - A11)$ and A_{21} is constructed as a factor of $(E + A_{11})$. k is a prime number that is mutually prime with R. When $A_{12} \neq 0$, formula (9) can be derived:

$$\begin{cases} A_{11} = -A_{22} \pmod{R}, \\ A_{12} = k(E - A_{11}) \pmod{R}, \\ A_{21} = \dfrac{(E + A_{11})}{k} \pmod{R}. \end{cases} \tag{9}$$

The self-invertible matrix A can be calculated using a given matrix A_{22}. Taking $k = 3$, $R = 256$ and $A_{22} = \begin{bmatrix} 215 & 52 \\ 20 & 45 \end{bmatrix}$ as an example, to calculate the self-reversible matrix A, first we should calculate the matrix A_{11}. Because $A_{11} = -A_{22} \pmod{R}$, we can get

$$A_{11} = \begin{bmatrix} -215 & -52 \\ -20 & -45 \end{bmatrix} \pmod{256} = \begin{bmatrix} 41 & 204 \\ 236 & 211 \end{bmatrix}. \tag{10}$$

Because $A_{12} = k(E - A_{11}) \pmod{R}$, we can get

$$A_{12} = 3 * (E - A_{11}) \pmod{256} = \begin{bmatrix} 136 & 156 \\ 60 & 138 \end{bmatrix}. \tag{11}$$

Because $A_{21} = (E + A_{11})/k \pmod{R}$, we can get

$$A_{21} = \frac{(E + A_{11})}{3} \pmod{256} = \begin{bmatrix} 14 & 68 \\ 164 & 156 \end{bmatrix}. \tag{12}$$

Then, the self-invertible matrix A can be obtained. $A = \begin{bmatrix} 41 & 204 & 136 & 156 \\ 236 & 211 & 60 & 138 \\ 14 & 68 & 215 & 52 \\ 164 & 156 & 20 & 45 \end{bmatrix}$. It can be verified that $A^{-1}A \pmod{256} = E$.

2.3. Fractal. In 1967, Mandelbrot published a paper entitled, "How Long is the British Coastline," in Science. In it, he used fractals to describe a large class of complex irregularities that cannot be described using traditional Euclidean geometry in nature. It marked the emergence of fractal thought. A fractal is a set of mathematical theories that uses fractal features as the research object. Some common geometric fractals are the Koch curve, the H-fractal, the Sierpinski triangle, and the Vivsek triangle. Fractal theory is not only a frontier and important branch of nonlinear science but also a new cross-discipline. It is a new mathematics discipline that studies the characteristics of a class of phenomena. Compared with its geometric form, it is more connected with differential equations and dynamic systems theory. The fractals are not limited to geometric forms and times and processes can also form fractals. As a new concept and method, the fractal is being applied in many fields. In recent years, fractal sensitivity, especially the sensitivity of the Mandelbrot sets and Julia sets to initial values, has been widely used in image encryption.

The H-fractal is a kind of fractal, and the diagram of the H-fractal is shown in Figure 2. Fractal graphics can be used for information encryption and security. This algorithm uses the H-fractal to encrypt image information.

3. Encryption Scheme

3.1. Key Generation. SHA-3 algorithm is a kind of Secure Hash Algorithm. This encryption algorithm uses the Hash sequence that is generated by the SHA-3(256) algorithm, and the prime number k is used to construct the self-invertible matrices that are used as keys. The initial values x_0, y_0, z_0, and w_0 of the Lorenz hyperchaotic system are generated by the original image. To obtain the 256 bit binary Hash sequence H, the algorithm inputs the original image into the SHA-3(256) function. Then, the sequence H is divided into 32 8 bit binary sequences as h_1, h_2, . . ., h_{32}, and

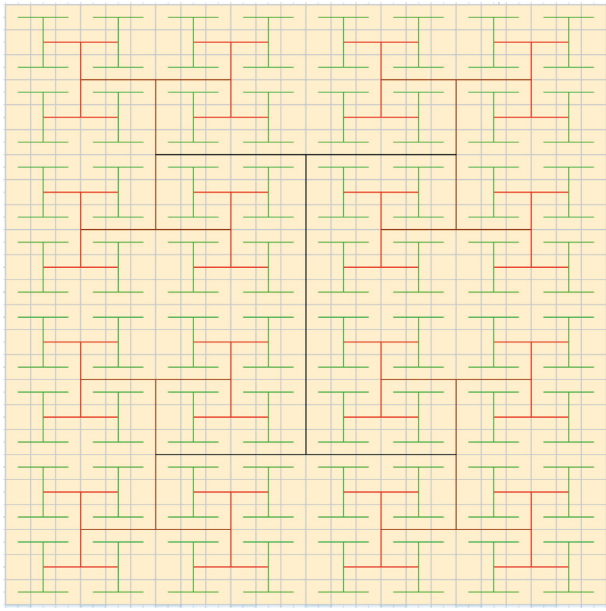

Figure 2: The diagram of the H-fractal.

the initial values of the Lorenz hyperchaotic system are calculated using

$$\begin{cases} x_0 = \dfrac{(h_1 \oplus h_2 \oplus h_3 \oplus h_4 \oplus h_5 \oplus h_6 \oplus h_7 \oplus h_8)}{256} + x_0', \\ y_0 = \dfrac{(h_9 \oplus h_{10} \oplus h_{11} \oplus h_{12} \oplus h_{13} \oplus h_{14} \oplus h_{15} \oplus h_{16})}{256} + y_0', \\ z_0 = \dfrac{(h_{17} \oplus h_{18} \oplus h_{19} \oplus h_{20} \oplus h_{21} \oplus h_{22} \oplus h_{23} \oplus h_{24})}{256} + z_0', \\ w_0 = \dfrac{(h_{25} \oplus h_{26} \oplus h_{27} \oplus h_{28} \oplus h_{29} \oplus h_{30} \oplus h_{31} \oplus h_{32})}{256} + w_0'. \end{cases} \tag{13}$$

The number of iterations of the hyperchaotic system is selected according to the size of the original image after obtaining the initial values of the Lorenz hyperchaotic system. If the size of the original image is $M \times N$, it is necessary to iterate the Lorenz hyperchaotic system $M \times N + 800$ times and delete the first 800 iterations to avoid the transient effect. Finally, four pseudorandom sequences X, Y, Z, and W of length $M \times N$ are obtained.

3.2. Scrambling Based on the Self-Invertible Matrix. The sequence Y that is generated by the Lorenz hyperchaotic system is chosen to produce the self-invertible matrices. When an $M \times N$ original image is encrypted using dynamic self-invertible matrices, the encryption process is described as follows:

Step 1: The original image is divided into $M \times N/16$ matrices that are sized 4×4, which are, respectively, labeled as PM_i ($i = 1, 2, \ldots, M \times N/16$) using the row-first method.

Step 2: The pseudorandom sequence Y in the Lorenz chaotic system is conducted using formula (14) to obtain the pseudorandom matrix YM. In formula (14), floor(x) is a floor function, and reshape(x) is a column-first ordering function:

$$YM = \text{reshape}((\text{mod}(\text{floor}((Y(:) * 10^2 - \text{floor}(Y(:) * 10^2)) * 10^{10}), 256), 256, 256)). \tag{14}$$

Step 3: The matrix YM is divided into $M \times N/16$ matrices that are sized 4×4 according to the method in Step 1, and we label the 4×4 matrices as YM_i ($i = 1, 2, \ldots, M \times N/16$).

Step 4: The 4×4 matrices YM_i ($i = 1, 2, \ldots, M \times N/16$) are divided into four 2×2 matrices, and the matrices in the upper left corner are reserved as YM_i' ($i = 1, 2, \ldots, M \times N/16$).

Step 5: The 2×2 matrices YM_i' ($i = 1, 2, \ldots, M \times N/16$) are transformed into the self-invertible matrices as K_i ($i = 1, 2, \ldots, M \times N/16$) using the prime number k with the construction method of the self-invertible matrix.

Step 6: The cipher matrices C_i ($i = 1, 2, \ldots, M \times N/16$) are calculated, and $C_i = K_i PM_i$ ($i = 1, 2, 3, \ldots, M \times N/16$).

Step 7: The cipher image is composed of the cipher matrices C_i ($i = 1, 2, \ldots, M \times N/16$) using the row-first method.

The decryption process of the self-invertible matrix encryption algorithm is the inverse of the encryption process, so it will not be described again.

3.3. H-Fractal Diffusion. The H-fractal cross-diffusion method that is proposed in this paper uses the intermediate pixel that is covered by the H-fractal as an operator to cross-process the two pixels on both ends of the H-fractal to complete the diffusion. Taking a 3×3 block as an example, the diffusion process based on the H-fractal is shown in Figure 3.

In Figure 3, pixel 2 is used as a control word, and pixel 1 and pixel 3 are controlled to perform a crossover operation. Then, pixel 8 is used as a control word, and pixel 7 and pixel 9 are controlled to perform a crossover operation. Finally, pixel 5 is used as a control word, and pixel 2 and pixel 8 are controlled to perform a crossover operation. The crossover operation method is shown in Figure 4, where E is the control word, B and D are the endpoint pixels, and B' and D' are the pixels after the crossover operation. When the values of the binary control bits in the pixel E are "1," the binary words in pixels B and D corresponding to the control bits are exchanged. Conversely, when the values of the binary control bits in pixel E are "0," the binary words in pixels B and D corresponding to the control bits have no operation. The decryption process is the inverse of the diffusion process, so it will not be described here.

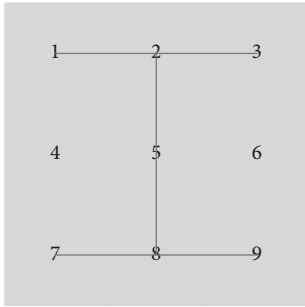

Figure 3: The diagram of H-fractal diffusion.

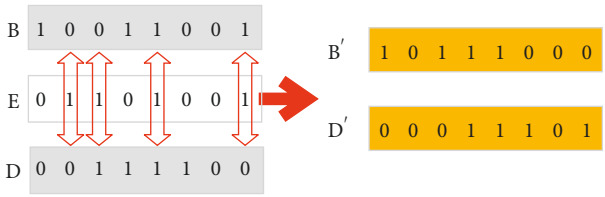

Figure 4: The diagram of the crossover operation.

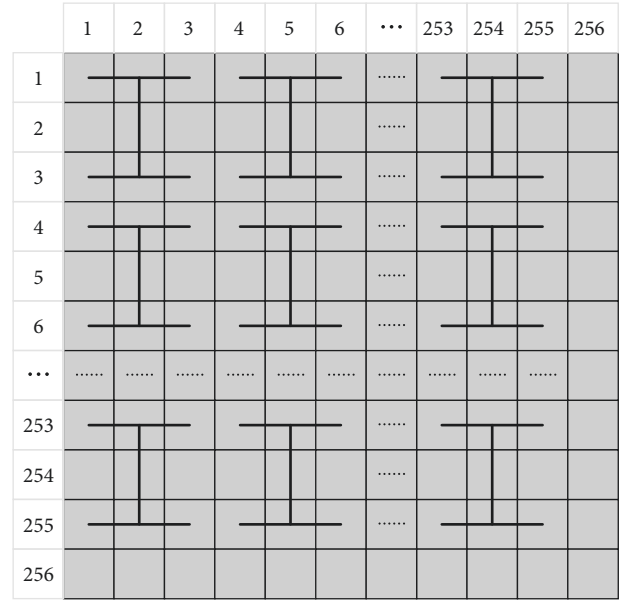

Figure 5: The image covered by the H-fractal.

Taking a 256×256 image as an example, the image that is covered by the H-fractal is shown in Figure 5. The first pixel in the upper left corner of the image is used as the starting point to construct the H-fractal. The pixels that are not covered by the H-fractal in the image are not operated on.

3.4. Cipher-Pixels Feedback Encryption. In this paper, a cipher-pixel feedback method is used to enhance the diffusion effect. The cipher-pixel feedback method makes the pixels in the front of the pixel sequence affect the pixels behind them. Assuming that the size of the original image is $M \times N$, the specific process of cipher-pixel feedback is described as follows. First, rearrange the original images into a pixel sequence $P\{1, 2, 3, \ldots, M \times N\}$. Then, a diffused sequence $P'\{1, 2, 3, \ldots, M \times N\}$ is obtained by operating on sequence P using

$$\begin{cases} P'(1) = P(1), \\ P'(i) = \text{bitxor}\left(P(i), P'(i-1)\right), \quad 2 \le i \le M \times N. \end{cases} \tag{15}$$

3.5. Scrambling Process. The algorithm uses the pseudorandom sequences X, Z, and W that are generated by the Lorenz hyperchaotic system to scramble the image. Taking the $M \times N$ image as an example, assuming that the given key is a pseudorandom sequence $S\{1, 2, 3, \ldots, M \times N\}$, the global scrambling operation is described as follows. First, the original image is expanded into a one-dimension pixel sequence $P_1\{1, 2, 3, \ldots, M \times N\}$, and the positions of the pixels in sequence P_1 are corresponded with the positions of the elements in sequence S. Then, the pseudorandom sequence S is rearranged in ascending order to obtain an index sequence S'. Finally, the pixel sequence $P_1\{1, 2, 3, \ldots, M \times N\}$ is mapped to the new pixel sequence $P_1'\{1, 2, 3, \ldots, M \times N\}$ according to the rules for mapping the elements in the sequence S to the sequence S'. The decryption process of scrambling is the inverse of the encryption process, so it will not be described here.

3.6. Encryption Scheme. The flow chart of the encryption scheme is shown in Figure 6. Taking image I that is sized 256×256 as an example, the encryption process is described as follows:

Step 1: Image $\mathbf{I}$ is input into the SHA-3(256) algorithm to obtain a Hash sequence H.

Step 2: The initial values x_0, y_0, z_0, and w_0 of the Lorenz hyperchaotic system are obtained by conducting the Hash sequence H.

Step 3: The Lorenz hyperchaotic system is iterated $M \times N + 800$ times, and four sequences X, Y, Z, and W are obtained by discarding the values of the first 800 iterations.

Step 4: The image matrix $\mathbf{I}_1$ is obtained by scrambling the original image $\mathbf{I}$ using sequence X.

Step 5: The image matrix $\mathbf{I}_2$ is obtained by using the dynamic self-invertible matrix encryption method to encrypt image matrix $\mathbf{I}_1$ using sequence Y.

Step 6: The image matrix $\mathbf{I}_3$ is obtained by scrambling the image matrix $\mathbf{I}_2$ using sequence Z.

Step 7: The image matrix $\mathbf{I}_4$ is obtained by using the H-fractal encryption method to encrypt image matrix $\mathbf{I}_3$.

Step 8: The image matrix $\mathbf{I}_5$ is obtained by scrambling the image matrix $\mathbf{I}_4$ using sequence W.

Step 9: The cipher-text feedback operation is performed on the image matrix $\mathbf{I}_5$ to obtain the image matrix $\mathbf{I}_6$, namely, the ciphertext image.

The decryption process of this encryption scheme is the inverse of the encryption process, so it will not be repeated.

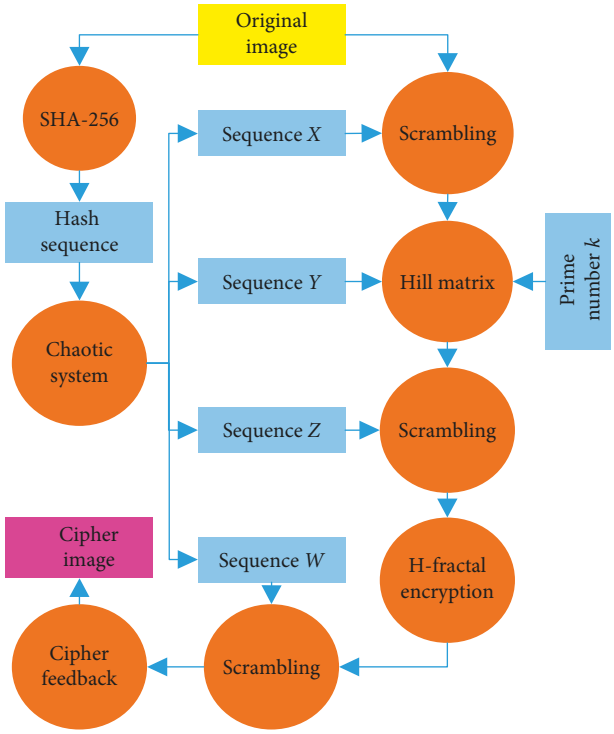

Figure 6: The flow chart of the encryption scheme.

4. Simulation Results and Security Analysis

In order to verify the effectiveness and feasibility of our algorithm, simulated experiments are undertaken on the MATLAB R2018a platform. The environment of development is Windows 7, 4.00 GB RAM, Intel(R) Core(TM) i3-4130 CPU @ 3.4 GHz. Mean execution time of test images with size of 256×256 is 1.518s. Part of the encryption keys are set as $x_0' = 0, y_0' = 0, z_0' = 0, w_0' = 0, a = 10,\ b = 8/3,\ c = 28, r = -1$ and the Hash sequences are generated by the original images. Some original images are shown in Figures 7(a)–7(d), and their cipher images are shown in Figures 7(e)–7(h). Because the encryption algorithm that we proposed is lossless, the decrypted images of the cipher images are exactly the same. The algorithm does not destroy the characteristics of the original image.

4.1. Key Sensitivity Analysis. The encryption algorithm that is proposed in this paper uses the 256 bit Hash sequence that is generated by the SHA-3(256) algorithm and the prime number k as the key. The key space of the 256 bit Hash sequence is 2^{128}. Therefore, the key space of the algorithm is large enough to resist brute-force attacks. The initial values of the Lorenz hyperchaotic system are generated by the Hash sequence. When the Hash sequence has a slight change, the initial value of the hyperchaotic system also changes. The algorithm is very sensitive to the changes of these initial values. When these initial values have been slightly changed by 10^{-13}, the encryption system cannot be decrypted. When the prime number $k = 3$, the original image and the correct decrypted image are shown in Figures 8(a) and 8(b). In Figures 8(c)–8(g), the decrypted images after the initial values have been changed are listed. The encryption algorithm is very sensitive to the key, and the algorithm is sufficient to resist attacks on the key.

4.2. Differential Attack Analysis. When the original image has a slight change, the cipher image will have a big change. This phenomenon reflects that the encryption system is very sensitive to changes in the original image. The higher the sensitivity of the plaintext, the stronger the cryptosystem's ability to resist differential attacks. Here, we use the number of pixel changes rate (NPCR) and the unified average changing intensity (UACI) to measure the antidifferential attack capability of the encryption system. The methods for calculating the NPCR and UACI are described as

$$\begin{cases} \mathrm{NPCR} = \dfrac{\sum_{i,j} \left| Sign\left(P_1(i,j) - P_2(i,j)\right) \right|}{M \times N} \times 100\%, \\ \mathrm{UACI} = \dfrac{\sum_{i,j} \left| P_1(i,j) - P_2(i,j) \right|}{255 \times M \times N} \times 100\%. \end{cases} \tag{16}$$

In formula (15), P_1 is the correct cipher image, and P_2 is the cipher image where the original image has a little change. M and N, respectively, represent the length and width of the image. Sign(x) represents the symbol function, and its calculation method is described as

$$\mathrm{Sign}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0. \end{cases} \tag{17}$$

The maximum theoretical value of the NPCR is 100%, and the ideal value of the UACI is 33.4635%. The larger the NPCR is, the greater the pixel changes. When the original image has been changed by 1 bit, the values of the NPCR and UACI are shown in Table 1. In addition, the values of the NPCR and UACI in the references [32] are listed in Table 1. By comparison, it is known that the algorithm that we proposed is very sensitive to plaintext and can resist differential attacks very well.

4.3. Information Entropy Analysis. Information entropy is the concept that was proposed by Shannon to quantify information. It can usually be expressed as $H(s)$. The concept of information entropy is described as

$$H(s) = -\sum_{i=1}^{n} p(i) \log_2 p(i). \tag{18}$$

In formula (17), $p(i)$ represents the probability of the occurrence of the case and n represents the total number of all possible occurrences. The information entropy is used to measure the randomness of the information. The closer the information entropy is to the ideal value, the stronger the randomness of the information is. The pixels in the grayscale image are all in the interval [0, 255]. When the image is completely random, the probability of each pixel value is 1/256, so the information entropy of a completely random grayscale image is 8. The information entropies of some

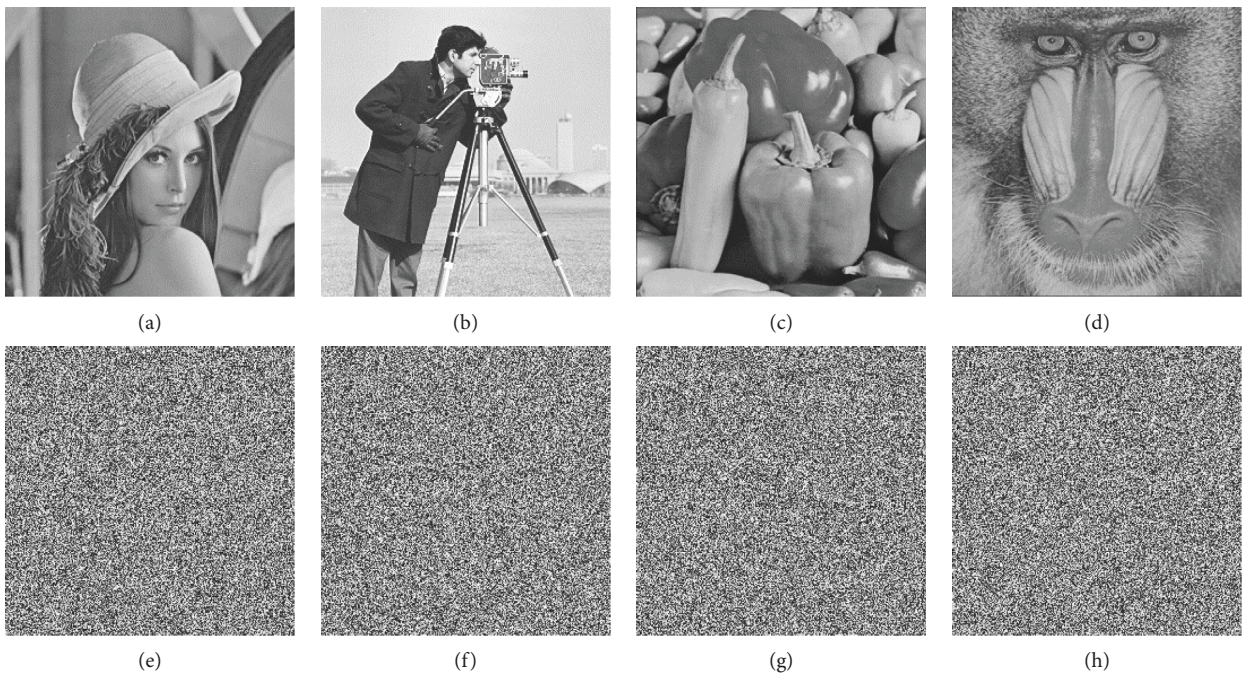

Figure 7: The original images and their cipher images. (a) Original Lena image. (b) Original Cameraman image. (c) Original Pepper image. (d) Original Baboon image. (e) Cipher Lena image. (f) Cipher Cameraman image. (g) Cipher Pepper image. (h) Cipher Baboon image.

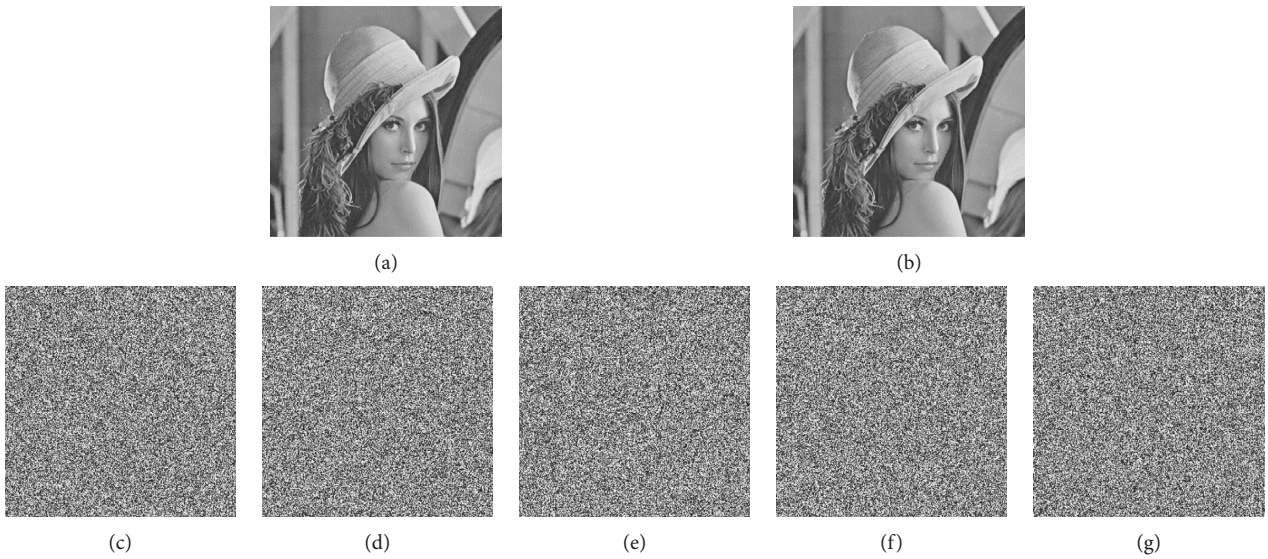

Figure 8: The correct decrypted image and the incorrectly decrypted images due to a slight change in the initial values of the Lorenz hyperchaotic system. (a) The original Lena image. (b) The correct decrypted image. The decrypted image (c) when x_0 is changed by 10^{-13}, (d) when y_0 is changed by 10^{-13}, (e) when z_0 is changed by 10^{-13}, (f) when w_0 is changed by 10^{-13}, and (g) when $k = 5$.

Table 1: NPCR and UACI.

Image	The proposed scheme		Reference [32]		Reference [33]	
	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Lena	99.6292	33.4481	99.5986	33.4561	99.58	33.08
Cameraman	99.6094	33.6017	99.5590	33.4439	99.90	33.15
Peppers	99.6216	33.5715	99.5803	33.4324	99.71	32.19
Baboon	99.6140	33.5152	—	—	99.59	31.56

Table 2: The information entropies of some original images and their cipher images.

Image	Entropy					
	Original	$k = 3$	$k = 37$	$k = 487$	Reference [32]	Reference [33]
Lena	7.4532	7.9971	7.9974	7.9974	7.9971	7.9968
Cameraman	6.9046	7.9976	7.9971	7.9972	7.9971	7.9904
Peppers	7.5797	7.9973	7.9978	7.9969	7.9968	7.9961
Baboon	7.0092	7.9972	7.9973	7.9975	—	7.9971

original images and their cipher images are listed in Table 2. It can be seen from the comparison that the cipher images that are encrypted by this algorithm are close to random.

4.4. Histogram Statistical Analysis. Histogram statistical analysis is a kind of statistical attack, and the histogram can characterize the image. The pixel distribution in the histogram of the original image is not uniform, which is not conducive to resisting statistical attacks. A good encryption algorithm can make the pixel distribution in the histogram of the cipher image more uniform, and thus, it can resist known-plaintext attacks and chosen-plaintext attacks. The histograms of the original images are shown in Figures 9(a)–9(c), and the histograms of the cipher images are shown in Figures 9(d)–9(f). It can be seen from the comparison that the algorithm can destroy the histogram of the statistical law of the original image and achieve good performance.

4.5. Correlation Analysis. 10000 pixels and their adjacent pixels from the original Lena image are randomly selected in the horizontal, vertical, and diagonal directions, and the values of these pixels are shown in Figures 10(a)–10(c), respectively. It can be seen from the analysis that there is a strong correlation between adjacent pixels. A good encryption algorithm can break the correlation between adjacent pixels, and it can enhance the ability to resist statistical attacks. 10000 pixels and their adjacent pixels from the cipher Lena image are randomly selected in the horizontal, vertical, and diagonal directions, and the values of these pixels are shown in Figures 10(d)–10(f), respectively. It can be seen from the comparison that the algorithm can break the correlation between adjacent pixels.

The correlation coefficient is used as an indicator to measure the correlation between adjacent pixels. Its calculation method is described as

$$\begin{cases} E(x) = \dfrac{1}{N}\sum_{i=1}^{N} x_i, \\ D(x) = \dfrac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)^2, \\ \operatorname{cov}(x, y) = \dfrac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)\left(y_i - E(y)\right), \\ r_{xy} = \dfrac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}. \end{cases} \tag{19}$$

In formula (18), N is the total number of selected pixels, $E(x)$ is the mean of the selected pixels, $D(x)$ represents the variance of the selected pixels, $\operatorname{cov}(x, y)$ represents the covariance of the selected pixels, and r represents the correlation coefficient. An absolute value of the correlation coefficient that is close to 1 indicates that the correlation of the data is strong, and an absolute value of the correlation coefficient that is close to 0 indicates that the data have almost no correlation. The correlation coefficients of the original images and the cipher images are listed in Table 3. It can be seen from the comparison that the image correlation coefficient of the encryption algorithm is almost zero, and the encryption algorithm can break the correlation between adjacent pixels.

4.6. Antiocclusion Attack Capability Analysis. The antiocclusion attack capability of the encryption system can reflect the degree of recovery of the decrypted image when the cipher image data are lost. In a cryptosystem without global scrambling, when the cipher image data are lost, its decrypted image may lose some important features in the original image. The Lena cipher images cut by 0, 1/256, 1/64, and 1/16 are shown in Figures 11(a)–11(d), and the corresponding decrypted images are shown in Figures 11(e)–11(h).

The NPCR, UACI, and correlation coefficients between the original images and the decrypted images after the occlusion attacks are listed in Table 4. The comparison between the data proves that the encryption algorithm that we proposed has a good antiocclusion attack capability.

4.7. Practicality Analysis. Some characteristics of the cipher images with different sizes encrypted by the proposed algorithm are listed in Table 5; these images are encrypted by $k = 3$. It can be seen from the data in Table 5 that the proposed algorithm can encrypt images with different size and has good encryption effects.

5. Conclusions

In this paper, an image encryption algorithm based on the H-fractal structure and dynamic self-invertible matrix is proposed. The algorithm uses the Hash sequence that is generated by the SHA-3(256) algorithm and a prime number as the keys. The image is scrambled and diffused by the four pseudorandom sequences that are generated by the Lorenz hyperchaotic system. In this encryption scheme, a cross-diffusion operation based on the H-fractal structure is applied for the first time. The algorithm enriches the means of

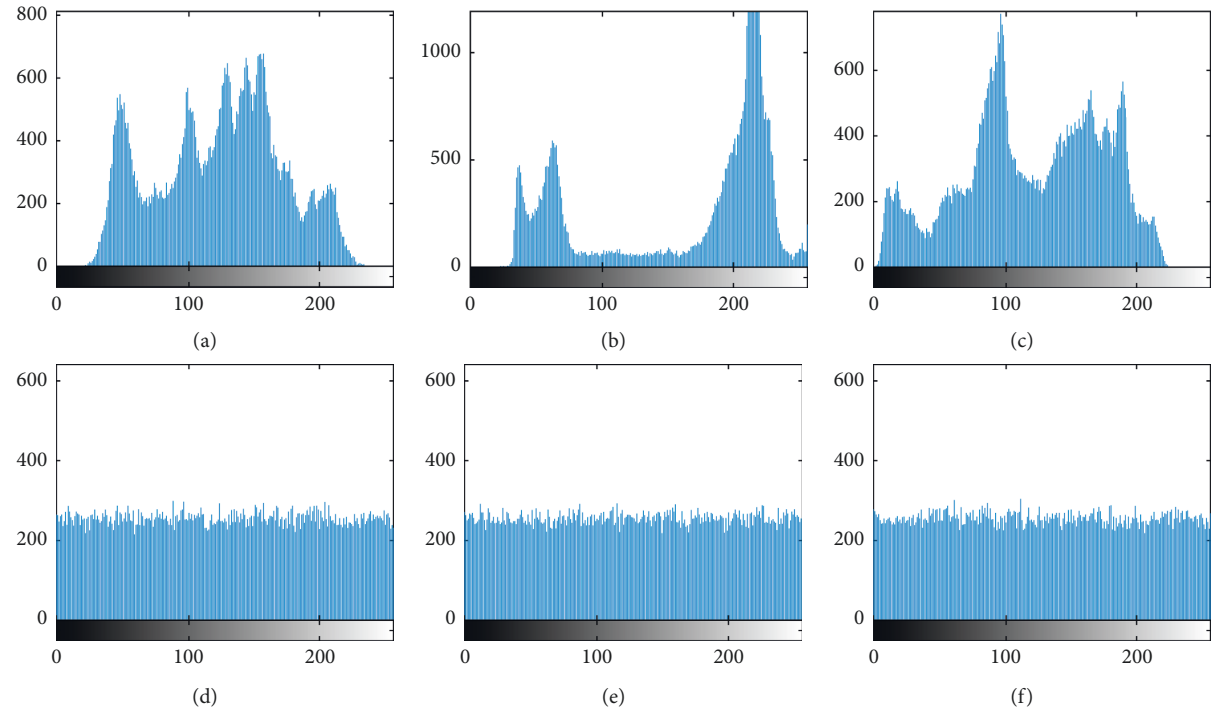

Figure 9: The histograms of the original images and their cipher images. The histograms of (a) the Lena image, (b) the Cameraman image, (c) the Pepper image, (d) the Lena cipher image, (e) the Cameraman cipher image, and (f) the Pepper cipher image.

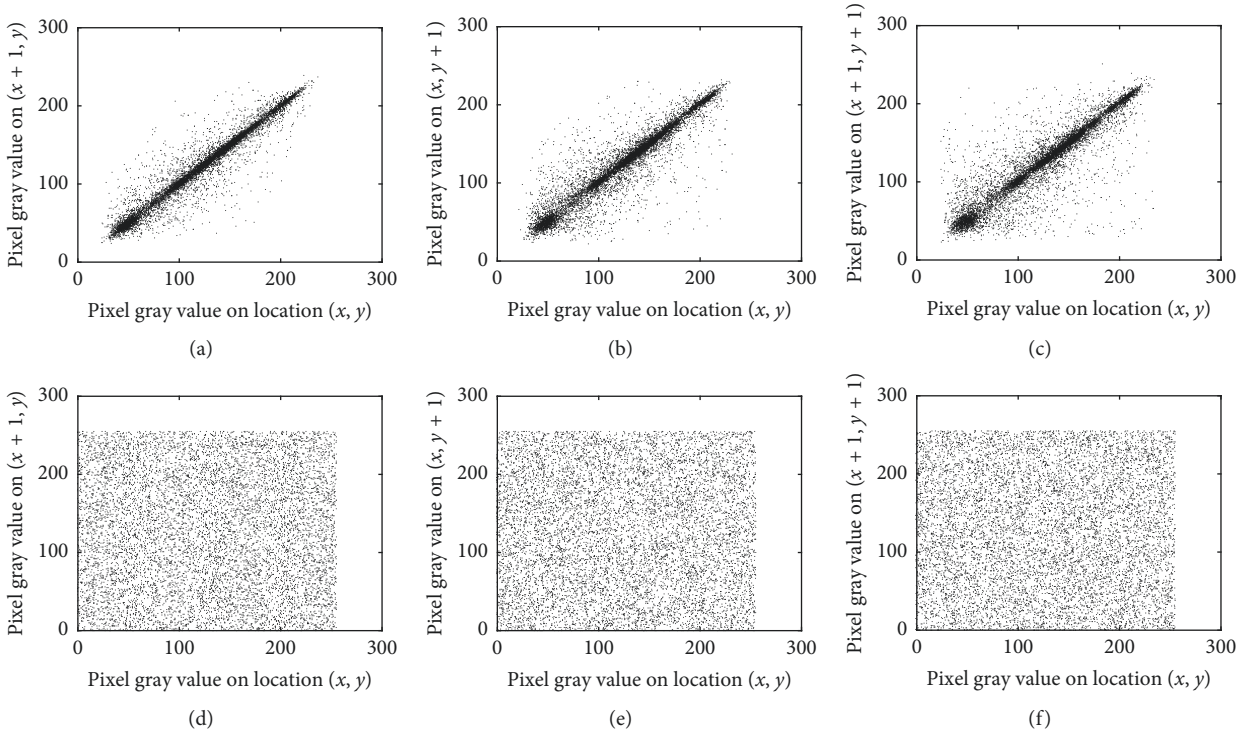

Figure 10: The values of the selected pixels and their adjacent pixels in different directions. Horizontal correlation of (a) the original image and (d) the cipher image. Vertical correlation of (b) the original image and (e) the cipher image. Diagonal correlation of (c) the original image and (f) of the cipher image.

Table 3: The correlation coefficients in different directions.

Image	Original image			Cipher image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	0.9680	0.9349	0.9069	0.0078	0.0040	−0.0050
Cameraman	0.9467	0.9180	0.9054	−0.0019	−0.0051	0.0032
Peppers	0.9731	0.9664	0.9381	0.0051	0.0037	0.0014
Baboon	0.8327	0.8759	0.7890	−0.0065	−0.0038	0.0065

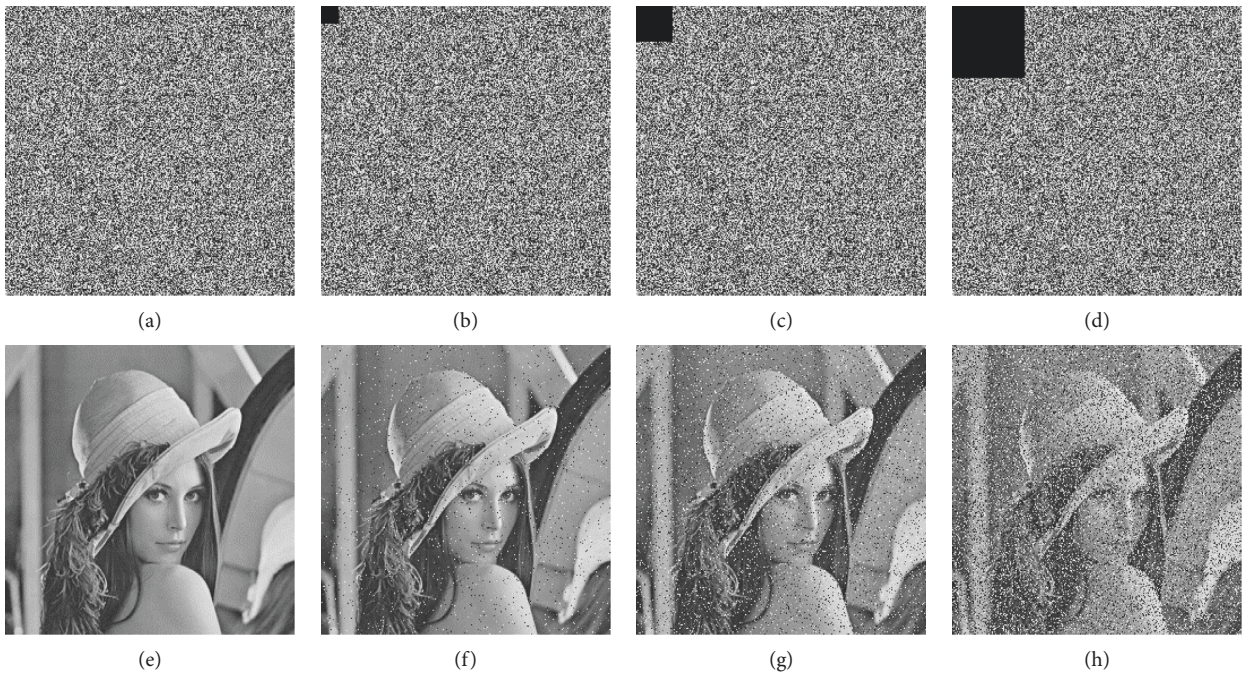

Figure 11: The reduced cipher images and their corresponding decrypted images. (a) Correct cipher image. (b) 1/256 occlusion. (c) 1/64 occlusion. (d) 1/16 occlusion. (e) The correct decrypted image. Decrypted image (f) with 1/256 occlusion, (g) with 1/64 occlusion, and (h) with 1/16 occlusion.

Table 4: The NPCRs, UACIs, and correlation coefficients of the images after the occlusion attack.

Occlusion	NPCR	UACI	Correlation coefficients		
			Horizontal	Vertical	Diagonal
0	0	0	0.9680	0.9349	0.9069
1/256	3.5339	1.0733	0.8371	0.8222	0.7957
1/64	12.2253	3.6316	0.6419	0.6057	0.5811
1/16	35.9364	10.6799	0.2669	0.2516	0.2327

Table 5: Analysis of cipher images with different sizes.

Cipher images	Correlation Coefficients			Entropies	NPCR (%)	UACI (%)
	Horizontal	Vertical	Diagonal			
Lena 128 × 128	0.0002	−0.0008	−0.0028	7.9882	99.6094	33.1566
Lena 512 × 512	−0.0061	0.0014	−0.0012	7.9994	99.6147	33.4730
Cameraman 128 × 128	0.0028	0.0020	0.0011	7.9878	99.6399	33.3091
Cameraman 512 × 512	0.0012	−0.0052	−0.0028	7.9993	99.5831	33.4335
Peppers 128 × 128	0.0028	−0.0078	0.0098	7.9887	99.6216	33.3656
Peppers 512 × 512	−0.0005	0.0018	0.0063	7.9992	99.5899	33.4390
Baboon 128 × 128	−0.0062	0.0012	0.0062	7.9869	99.7253	33.6223
Baboon 512 × 512	−0.0071	−0.0053	−0.0067	7.9993	99.6162	33.5096

digital image encryption. It has high security to resist brute-force attacks and statistical attacks, and it has the ability to recover when the cipher data are lost. Thus, this algorithm can be used to protect the security of digital images.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (grant nos. 61572446, 61602424, and U1804262), Key Scientific and Technological Project of Henan Province (grant nos. 174100510009 and 192102210134), and Key Scientific Research Projects of Henan High Educational Institution (18A510020).

References

[1] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[2] D. Coppersmith, D. B. Johnson, and S. M. Matyas, "A proposed mode for triple-DES encryption," *IBM Journal of Research and Development*, vol. 40, no. 2, pp. 253–262, 1996.

[3] S. Kumar and S. Srivastava, "Image encryption using simplified data encryption standard (S-DES)," *International Journal of Computer Applications*, vol. 104, no. 2, pp. 38–42, 2014.

[4] G. Lokeshwari, S. Susarla, and S. U. Kumar, "A modified technique for reliable image encryption method using merkle–hellman cryptosystem and RSA algorithm," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 18, no. 3, pp. 293–300, 2015.

[5] P. Irfan, Y. Prayudi, and I. Riadi, "Image encryption using combination of chaotic system and rivers shamir adleman (RSA)," *International Journal of Computer Applications*, vol. 123, no. 6, pp. 11–16, 2015.

[6] Z. Kai, Z. Lin, and D. Yun, "A new RSA image encryption algorithm based on singular value decomposition," *International Journal of Pattern Recognition & Artificial Intelligence*, vol. 33, no. 1, article 1954002, 2018.

[7] B. Kumar Singh and S. Kumar Gupta, "Grid-based image encryption using RSA," *International Journal of Computer Applications*, vol. 115, no. 1, pp. 26–29, 2015.

[8] S. S. Yim, S. J. An, M.-J. Han, J. W. Choi, and K. J. Jeong, "Isolation of a potential anchoring motif based on proteome analysis of *Escherichia coli* and its use for cell surface display," *Applied Biochemistry and Biotechnology*, vol. 170, no. 4, pp. 787–804, 2013.

[9] A. Kannammal and S. Subha Rani, "DICOM image authentication and encryption based on RSA and AES algorithms," *Communications in Computer and Information Science*, vol. 330, pp. 349–360, 2012.

[10] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Information Sciences*, vol. 181, no. 6, pp. 1171–1186, 2011.

[11] 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.

[12] W. Y. Ji and H. Kim, "An image encryption scheme with A pseudorandom permutation based on chaotic maps," *Communications in Nonlinear Science & Numerical Simulation*, vol. 15, no. 12, pp. 3998–4006, 2010.

[13] C. Fu, B.-B. Lin, Y.-S. Miao, X. Liu, and J.-J. Chen, "A novel chaos-based bit-level permutation scheme for digital image encryption," *Optics Communications*, vol. 284, no. 23, pp. 5415–5423, 2011.

[14] V. Patidar, N. K. Pareek, G. Purohit, and K. K. Sud, "A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption," *Optics Communications*, vol. 284, no. 19, pp. 4331–4339, 2011.

[15] C. Fu, J.-J. Chen, H. Zou, W.-H. Meng, Y.-F. Zhan, and Y.-W. Yu, "A chaos-based digital image encryption scheme with an improved diffusion strategy," *Optics Express*, vol. 20, no. 3, pp. 2363–2378, 2012.

[16] K. W. Wong, S. H. Kwok, and C. H. Yuen, "An efficient diffusion approach for chaos-based image encryption," *Chaos Solitons & Fractals*, vol. 41, no. 5, pp. 2652–2663, 2009.

[17] W. Zhang, K.-W. Wong, H. Yu, and Z.-L. Zhu, "An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 8, pp. 2066–2080, 2013.

[18] X.-J. Tong, "The novel bilateral—diffusion image encryption algorithm with dynamical compound chaos," *Journal of Systems and Software*, vol. 85, no. 4, pp. 850–858, 2012.

[19] N. Benyamin, M. Sattar, S. S. Mohammad, and M. M. Reza, "A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process," *Multimedia Tools & Applications*, vol. 71, no. 3, pp. 1469–1497, 2014.

[20] N. Bourbakis and C. Alexopoulos, "Picture data encryption using scan patterns," *Pattern Recognition*, vol. 25, no. 6, pp. 567–581, 1992.

[21] S. S. Maniccam and N. G. Bourbakis, "Image and video encryption using SCAN patterns," *Pattern Recognition*, vol. 37, no. 4, pp. 725–737, 2004.

[22] A. Jolfaei and A. Mirghadri, "An image encryption approach using chaos and stream cipher," *Journal of Theoretical & Applied Information Technology*, vol. 19, no. 2, pp. 117–125, 2010.

[23] B. Acharya, S. K. Panigrahy, S. K. Patra, and G. Panda, "Image encryption using advanced hill cipher algorithm," *International Journal of Recent Trends in Engineering*, vol. 1, no. 1, pp. 663–667, 2009.

[24] S. F. Elzoghdy, Y. A. Nada, and A. A. Abdo, "How good is the DES algorithm in image ciphering?," *International Journal of Advanced Networking & Applications*, vol. 2, no. 5, pp. 796–803, 2011.

[25] A. Gehani, T. Labeanv, and J. Reif, "DNA-based cryptography," *Aspects of Molecular Computing*, vol. 54, no. 456, pp. 233–249, 2004.

[26] Z.-H. Guan, F. Huang, and W. Guan, "Chaos-based image encryption algorithm," *Physics Letters A*, vol. 346, no. 1–3, pp. 153–157, 2005.

[27] X. Tong and M. Cui, "Image encryption with compound chaotic sequence cipher shifting dynamically," *Image and Vision Computing*, vol. 26, no. 6, pp. 843–850, 2008.

[28] X.-J. Tong, M. Zhang, Z. Wang, and J. Ma, "A joint color image encryption and compression scheme based on hyperchaotic system," *Nonlinear Dynamics*, vol. 84, no. 4, pp. 2333–2356, 2016.

[29] Q. Zhang and X. P. Wei, "A novel image fusion encryption algorithm based on DNA sequence operation and hyperchaotic system," *International Journal for Light and Electron Optics*, vol. 124, no. 23, pp. 6276–6281, 2014.

[30] S. C. Qu, D. Liu, and L. Wang, "Synchronization of hyperchaotic Lorenz system and its application in secure communication," *Key Engineering Materials*, vol. 467–469, pp. 437–440, 2011.

[31] D. H. Ou, W. Sun, and B. Lin, "A novel image encryption scheme with the capability of checking integrity based on inverse matrix," *Journal of Graphics*, vol. 33, no. 2, pp. 89–92, 2012.

[32] X. Wang, X. Zhu, and Y. Zhang, "An image encryption algorithm based on josephus traversing and mixed chaotic map," *IEEE Access*, vol. 6, pp. 23733–23746, 2018.

[33] S. Mozaffari, "Parallel image encryption with bitplane decomposition and genetic algorithm," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25799–25819, 2018.