Hindawi
Security and Communication Networks
Volume 2022, Article ID 3664986, 11 pages
https://doi.org/10.1155/2022/3664986

Research Article

Secure Analysis for IIOT Systems Using Hyperchaotic Image Encryption

Haini Zeng and Qiping Zou

School of Artificial Intelligence and Smart Manufacturing, Hechi Univesity, Yizhou 546300, China

Correspondence should be addressed to Qiping Zou; 706611232@qq.com

Received 8 October 2021; Revised 8 April 2022; Accepted 29 April 2022; Published 24 June 2022

Academic Editor: Xiaolong Xu

Like edge computing, intelligent cameras and image sensors are widely used in the Industrial Internet of Things (IIOT), including design and finished product quality inspection. However, the images generated by these sensors are constantly at risk of information leakage and privacy violations in the IIOT. Due to the involvement of third parties, traditional encryption algorithms are no longer adapted to image encryption for IIOT. In the context of the IIOT, an image encryption technology based on hyperchaotic systems and dynamic DNA coding is proposed. First, the image pixel position is scrambled by the hyperchaotic mapping index sequence, so that the image pixel matrix is dynamically DNA coded, and the base operation is performed on the given DNA sequence. Then, Keccak is used to calculate the hash value of the given DNA sequence as the initial value of the chaotic system and a certain number of base substitutions are performed on the DNA encoded pixel value according to the quaternary hyperchaotic sequence generated by the hyperchaotic system. Finally, ciphertext feedback and chaotic system iteration are used to further enhance the confusion and diffusion characteristics of the algorithm. The test results show that the algorithm not only has a large key space, strong sensitivity to keys, but also has strong resistance to exhaustive analysis attacks.

1. Introduction

As one of the most promising industries in the world today, the Internet of Things (IOT) is strongly driving the digital transformation and upgrading of traditional industries, enabling human society to enter the era of the Internet of everything [1–6]. The IOT collects, perceives, and analyzes the corresponding data from the surrounding environment through connected nodes and makes specific responses. Edge computing is widely used in IIOT to make up for the deficiencies of cloud computing [7–9]. Wireless multimedia sensor networks (WMSNs) is one of the IOT auxiliary devices, which consists of vision sensors. The WMSNs supervises the surrounding environment by continuously capturing images of the surrounding environment by visual sensors. However, the large amount of visual data obtained has significant redundancy [10]. Researchers of surveillance networks generally agree that multimedia surveillance networks should have visual data collection and record sensitive data for future use, such as anomalous event detection, case management, data analysis, and video abstraction. Due to energy and bandwidth constraints, it is impractical to send unprocessed video data over communication lines. In addition, extracting sensitive data from the large amount of surveillance data is difficult and time-consuming [11]. Therefore, it is necessary to exploit the processing and transmission capabilities of smart vision sensors to autonomously collect important visual data. This facilitates the intelligent selection of the appropriate picture from the multiview surveillance data captured by the connected IOT infrastructure of multiple sensors. It can process the collected data in real time, in order to send relevant data to a central memory. In addition, it enables surveillance experts to grasp the relevance of the original lengthy sequence by simply analyzing representative frames. However, when visualization data from WMSNs are sent wirelessly to a vision processing center (VPH) or a base station (BS), the communication is vulnerable to several security issues.

Therefore, some security mechanisms need to be designed to transmit the visual data safely to the BS because any slight change in the transmitted data can affect the decision of visual data analyst at the base station. In addition, it is comparatively difficult to transmit multimedia data in WMSNs using dedicated lines due to congestion in the bandwidth allocation mechanism.

Chaotic mappings have been widely used in the design of encryption schemes due to their unpredictability, ergodicity, and sensitivity to parameters and initial values [12]. It is generally more common to use chaotic mappings for the generation of pseudorandom number sequences, but studies have shown that the dynamical properties of chaotic mappings degrade to some extent. When calculating chaotic mapping values, the limitation of accuracy makes the chaotic mapping exist with finite and periodic orbits [13], which leads to degradation of all properties of the chaotic system and makes the chaos-based encryption scheme security flawed. Therefore, it is necessary to investigate the dynamical properties of chaotic mappings with finite accuracy. The structure and properties of several chaotic mappings have been studied by some scholars. In 2016, Yoshioka and Kawano analyzed in detail the relationship between the period, initial value, and order of Chebyshev polynomials on the ring of integer power remainders of 2 [14]. In 2019, Li et al. derived a strong correlation between the nodes corresponding to one-dimensional chaotic mappings on the domain of fixed-point operators and proved the number of iterations required for the iteration value of the tent mapping to converge to zero [15].

Due to the large amount of image data and high redundancy, in order to better meet the requirements of image protection, some researchers combine DNA encoding rules and chaotic mapping theory to propose some new image encryption methods [16–19]. For example, Chai et al. [17, 18] established an image encryption algorithm based on chaos by making appropriate improvements. Encryption is realized by combining with DNA code, and it turns out that its encryption performance reaches a higher level and the efficiency is high. Only DNA coding rules are used in the algorithm, which is simple to operate and shows strong applicability. However, in practical applications, most of the DNA encoding rules selected are fixed, which makes the algorithm's ability to resist exhaustive attacks very weak, so this is likely to cause security risks.

Therefore, this paper addresses these issues by employing an intelligent and efficient system that intelligently collects important data and gives appropriate decisions in real time through each sensor node, thus reducing bandwidth consumption and transmission costs. In addition, this paper proposes a security algorithm for secure transmission of sensitive visual data to the fusion center. We combine the hyperchaotic system, DNA calculation, and Keccak function to encrypt the image in chunks. The hash function Keccak processes is used to process the original image to obtain the initial value of the hyperchaotic system. Technically, the system encrypts the visual data using image encryption before transmitting the data, thus improving the security during communication in industrial WMSNs.

The approach and results show that the proposed encryption algorithm can encrypt different images securely and efficiently, making it more suitable for Industrial Internet of Things.

The rest of the paper is organized as follows: Section 2 shows the proposed system in detail. Section 3 shows encryption algorithm design. Section 4 presents the experimental results, and then the study concludes in Section 5.

2. Hyperchaotic System and Dynamic DNA Coding Algorithm

In the process of obfuscation and diffusion operation, the encryption algorithm in this article uses two kinds of chaotic sequences, DNA sequence library and pixel gray value conversion operation to achieve the purpose of encryption.

2.1. Improved Hyperchaotic Systems. Chaos is a complex phenomenon in nature. In 1963, Lorenz [20–25] used computer numerical experiments to discover the first chaotic attractor. This discovery is an important milestone in the study of chaos. Since then, the study of chaos has permeated almost all fields of natural science and social science. In 1979, Rössler discovered the first four-dimensional hyperchaotic system, i.e., the Rössler hyperchaotic system [21]. Compared with chaotic phenomena, hyperchaos expands in two or more directions and has at least two positive Lyapunov exponents. So the lowest dimension of hyperchaotic system is four, and there is at least one nonlinear term. This makes the hyperchaotic system present more complex dynamics, showing stronger randomness and unpredictability.

Zhang et al. [22] proposed a three-dimensional continuous autonomous chaotic system with the equation of state shown in the following equation:

$$\begin{cases} \dot{x} = \alpha x + yz \\ \dot{y} = -x + cy \\ \dot{z} = dy^2 - bz. \end{cases} \tag{1}$$

In equation (1), x, y, and z are system state variables and α, b, c, and d are system real parameters. When $a = 20, b = 5, c = 10$, $an\ d\ d = 7$, the system can produce chaotic attractors, and the three Lyapunov exponents of the system are $L_1 = 1.2371, L_2 = -0.0291$, and $L_3 = -16.4484$. The structure of the system formed under such conditions is more complex than that of the low-dimensional system, which can produce chaotic sequences of certain combinatorial forms. It makes the design of sequences more flexible and can better meet the application requirements. Given the initial values in the specific application, the corresponding sequences x, y, and z can be determined. The three are arranged in ascending order to determine x', y', and z'. The set of replacement addresses, corresponding to X, Y, and Z, is obtained by performing a certain position comparison. The image pixel position matrix can be scrambled by this sequence in a specific application. The 3 index sequences are determined and some scrambling operations are performed by them.

Based on this system, a new four-dimensional hyperchaotic system is constructed by first modifying three equations, followed by introducing a fourth-dimensional state variable using the state feedback control method and the second equation used for the introduced variable. Its state equation is shown in the following equation:

$$\begin{cases} \dot{x} = a(y - x) + bw \\ \dot{y} = -cx + 2\ dy\ z \\ \dot{z} = h - fy^2 - ksin(z) \\ \dot{w} = gx. \end{cases} \tag{2}$$

In equation (2), c and g are nonzero real numbers. x, y, z, and w are the state variables of the system. a, b, c, d, and r are the control parameters of the system. When $(a, b, c, d, g, f, h,$ and $k)$ is equal to $(15, 3, 8, 8, 2.7, 1, 3,$ and $2)$, respectively, the system behaves as hyperchaotic motion. The corresponding Lyapunov exponents are $L_1 = 0.6307, L_2 = 0.2868, L_3 = 0.0001,$ and $L_4 = -9.0857$. One of the methods and criteria for determining chaos is the Lyapunov exponent. If a positive Lyapunov exponent is obtained, the system is determined to be chaotic. If more than one positive Lyapunov exponent is obtained, the system is determined to be hyperchaotic.

When $(a, b, c, d, g, f, h,$ and $k)$ is equal to $(15, 3, 8, 8, 2.7, 1, 3,$ and $2)$, respectively, the system can generate topologically complex hyperchaotic attractors. The kinetic equations of equation (2) are calculated using the built-in Runge–Kutta function (ode45) in Matlab2020b and solved to obtain four one-dimensional chaotic sequences. Figures 1(a)–1(d) show the four chaotic attractor phase diagrams of *x-y-z*, *x-y-h*, *x-z-h*, and *y-z-h* for the last 40,000 data from each of the four sets of data obtained by using ode45 to calculate the hyperchaotic system in this paper.

From Figure 1, the system has the following characteristics: (a) The system has high dimensionality and can generate four different chaotic sequences. (b) The system structure is complex and the generated chaotic sequences have higher entropy values. (c) The four initial values of the system greatly affect the generated chaotic sequences, and all four initial values can be used as keys, which increase the key capacity and the difficulty of breaking the system.

2.2. Keccak Algorithm. Keccak [26, 27] is a standard one-way hash function algorithm. NIST's evaluation of Keccak is that the algorithm has very good security and implementation. Especially, it is designed in a completely different way compared to SHA-2, avoiding many known attacks and providing some performance that SHA-2 does not have. Keccak can generate hash values of any length, but to match the SHA-2 hash length, the SHA-3 standard specifies four output length versions: 224 bit, 256 bit, 384 bit, and 512 bit. In terms of the maximum length of the input data, SHA-3 is $2^{64} - 1$ bits, SHA-2 is $2^{128} - 1$ bits, and SHA-3 has no length limit. Keccak uses a sponge construction which is completely different from the SHA-1 and SHA-2 algorithms. In sponge construction, after the input data is filled, it goes through an absorbing phase and a squeezing phase to generate the output hash. The hash function can calculate a fixed-length hash value based on a message of any length. Attaching the hash value to the message or storing it together with the message can prevent the message from being modified during storage or transmission. Different messages have different hash values. As long as one bit changes in the message, the hash value will be completely different. Using this feature, by selecting the appropriate message, the hash value generated by the Keccak hash function is used to perform operations on the image to change the pixel value of the image. At the same time, the hash value is modified to set the initial value and system parameters of the chaotic system, so as to further improve the security of encryption. Keccak has no limit on the upper limit of the length of the input data and can generate any degree of hash value.

After the original image is converted with Keccak, a set of 512-bit hash values will be generated: 9caa44db566cfe1-f6a98c4991fffe891bb7d7fdf840449a026e923e9feab60b8b7e-d7a3933a757358c2c9441366976fab4bda222f9b5e4d-f814322e0dc12c13f. The generated hash values are used as input information for the next hash function to generate new hash values. The cycle is generated eight times to obtain a total of 256×8 bit hash values. A DNA encoding rule is chosen to encode the obtained hash values, and every 8-bit group of hash value is encoded to convert the 256×8 bit hash values into a 16×16 DNA encoding matrix. For example, according to the first encoding rule: db $\longrightarrow$ 11011011 $\longrightarrow$ TGCT.

In this article, the hash value K is generated by the Keccak algorithm, and then the initial value of the chaotic system is generated. Dividing K by bytes, it can be expressed as $k_1, k_2, k_3, \cdots k_{64}$. The initial value of the hyperchaotic system is calculated by the following formulae:

$$h_j = \frac{\left(k_{j+1} \oplus k_{j+2} \oplus k_{j+3}\right) + k_{j+4} + k_{j+5} + k_{j+6}}{256}, \tag{3}$$

$$\begin{cases} x_0 = 1 + abs\left(roun\ d\left(h_1\right) - h_1\right), \\ y_0 = 1 + abs\left(roun\ d\left(h_2\right) - h_2\right), \\ z_0 = 1 + abs\left(roun\ d\left(h_3\right) - h_3\right), \\ w_0 = 1 + abs\left(roun\ d\left(h_4\right) - h_4\right). \end{cases} \tag{4}$$

Among them, $j = 6(i - 1)$, where $i = 1$, 2, 3, and 4. The keys generated in this way have the advantages of good randomness, periodicity, and long key space properties. By combining the original image information with the key, the algorithm will effectively resist known plaintext and selected plaintext attacks.

2.3. Dynamic DNA Coding Technology. At present, the scale of nucleic acid databases has increased substantially, and the corresponding growth rate can be described by an exponential law. The corresponding data capacity is very large and can be regarded as a natural code book. DNA sequence [11] is mainly used for ciphertext diffusion and hash value generation. The DNA molecule consists of adenine (A), cytosine (C), guanine (G), and thymine (T). A specific analysis shows that a DNA molecule can be formed by

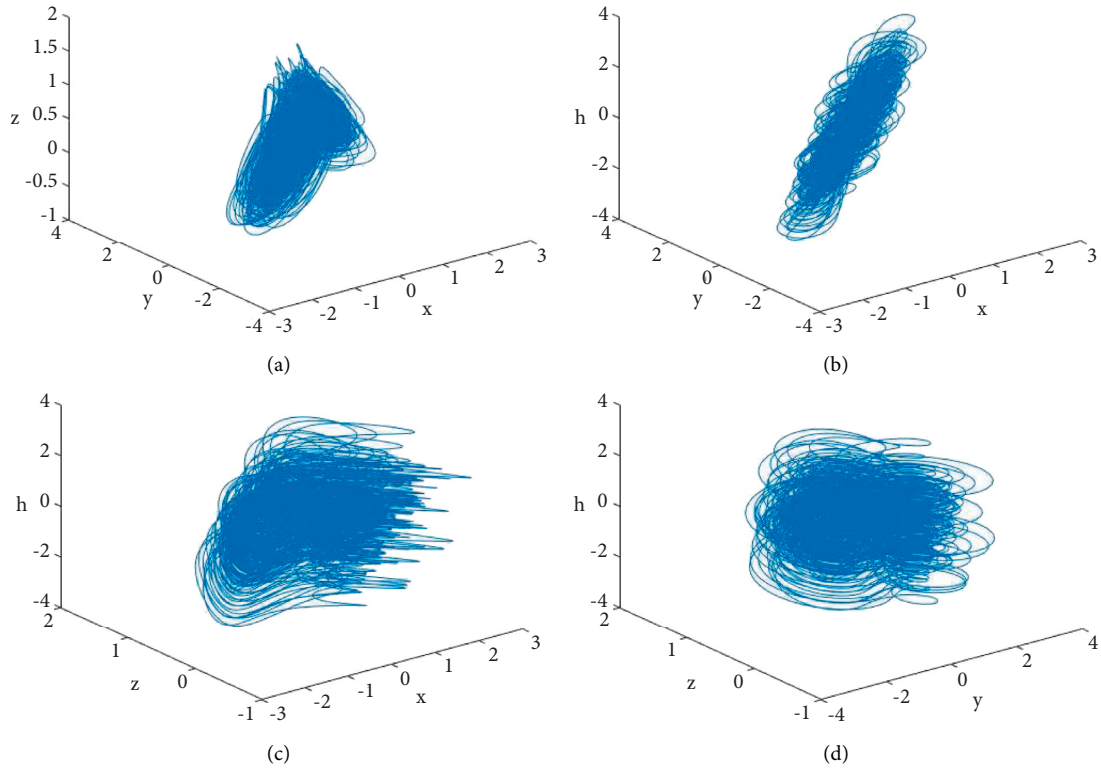

Figure 1: Phase diagram of the new four-dimensional hyperchaotic system. (a) *x-y-z* chaos attractor phase diagram, (b) *x-y-h* chaos attractor phase diagram, (c) *x-z-h* chaos attractor phase diagram, and (d) *z-y-h* chaos attractor phase diagram.

binding two single-stranded DNA molecules under the action of hydrogen bonds. The principle of complementary base pairing in this binding process is related to the base characteristics, and the corresponding pairing rules are expressed as hydrogen bond pairing of A and T and hydrogen bond pairing of G and C [11]. This combination of characteristics is similar to the binary formed by semiconductor pass-throughs. Thus, it is possible to store and process information on the basis of such combinations and meet certain requirements for computational analysis [25, 28].

2.3.1. Coding Rules. If we proceed according to the A ⟶ 00, B ⟶ 01, C ⟶ 10, and T ⟶ 11 rule, we match the complementary pairing A–T and C–G of the base pair. The relevant complementary pairing rules in this case are specified in Table 1.

The grayscale value of each pixel of the grayscale image is described by the corresponding 8-bit binary number. In the case of DNA encoding, a simple 4-base sequence is encoded and then converted into a DNA sequence, and the conversion rules for the DNA sequence are used in the image processing. In the encrypted image, the following base substitution rules are set to meet the interference requirements and to improve the confidentiality.

2.3.2. Base Substitution Rules. Setting a specific mapping function $L(x)$, the following relational rules are determined.

Table 1: Coding rules.

Rule	1	2	3	4	5	6	7	8
00	A	A	C	G	C	G	T	T
01	C	G	A	A	T	T	C	G
10	G	C	T	T	A	A	G	C
11	T	T	G	C	G	C	A	A

$$\begin{cases} x \neq L(x) \neq L(L(x)) \neq L(L(L(x))) \\ x = L(L(L(L(x)))). \end{cases} \tag{5}$$

Here, $x \in \{A, C, G, T\}$, there are six reasonable combinations of base substitutions according to this convention.

The permutation process of the images can be scrambled by randomly selecting any of the permutation combinations in Table 2 according to the application requirements, based on which the encoding is performed.

2.3.3. Base Algebraic Operation Rules. A ⟶ 00, C ⟶ 01, G ⟶ 10, and T ⟶ 11 codes are set according to the complementary pairing rules. Exclusive OR, addition, and subtraction rules of the DNA are shown in Tables 3–5, respectively. For other codes, similar operation rules are determined on these basis.

In this article, Keccak algorithm is used to generate hash value K for DNA sequence, and the length of K is 512 bits. The gray value diffusion is used to process the image pixel gray value and DNA sequence by base operation. During this

Table 2: Base substitution rules.

1	A ⟶ T ⟶ C ⟶ G ⟶ A
2	A ⟶ T ⟶ G ⟶ C ⟶ A
3	A ⟶ C ⟶ T ⟶ G ⟶ A
4	A ⟶ C ⟶ G ⟶ T ⟶ A
5	A ⟶ G ⟶ T ⟶ C ⟶ A
6	A ⟶ G ⟶ C ⟶ T ⟶ A

Table 3: Exclusive OR operation rules.

XOR	A	C	G	T
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A

Table 4: Addition operation rules.

XOR	A	C	G	T
A	A	C	G	T
C	C	G	T	A
G	G	T	A	C
T	T	A	C	G

Table 5: Subtraction operation rules.

XOR	A	C	G	T
A	A	T	G	C
C	C	A	T	G
G	G	C	A	T
T	T	G	C	A

operation, the starting base position R of the sequence must be set. The dynamic DNA coding technology is mainly based on the analysis of the position in the pixel matrix and the hash value K in the coding process, and then the appropriate coding rules are determined.

The corresponding DNA coding rules are as follows:

$$R_{i,j} = Mo\ d((i-1)*N+j;8) \oplus Bin2\ de\ c\left(k_s k_{s+1} k_{s+2}\right). \quad (6)$$

Among them, $i \in \{1, 2, \ldots, M\}$, $j \in \{1, 2, \ldots, N\}$, and $s = Mo\ d((i-1)*N+j-1, 510)+1$. $K_s K_{s+1} K_{s+2}$ are composed of three binary bits of s bit, $s+1$ bit, and $s+2$ bit, respectively, of the hash value K. Since each pixel value of the image can be represented by 8-bit binary, and each pixel corresponds to 4 bases, it can be determined that the length of the DNA_S after the encoding process is $4 \times M \times N$.

3. Encryption Algorithm Design

The algorithm in this article is divided into two parts. First, pixel position scrambling transformation is performed. In the operation process, a permutation index is constructed based on the Lorentz-chaotic sequence to scramble the image. Each pixel of the original image is converted into DNA sequence information, and ciphertext feedback processing is performed according to a certain sequence to achieve the purpose of replacement. The encryption flowchart is shown in Figure 2, and the details of the encryption process are as follows:

Step 1: input the gray image I and determine the output size as two-dimensional matrix $I_1 = M \times N$.

Step 2: obtain the index sequence X from formula (1) and scramble the matrix I_1 to obtain a new matrix I_2.

Step 3: use dynamic DNA coding to process I_2 and obtain a new DNA coding matrix I_3.

Step 4: download the DNA sequence of the gene bank, and intercept $4 \times M \times N$ base sequences from R to form a matrix I'.

Step 5: XOR the base sequences corresponding to I_3 and I' to obtain a matrix I_4 and scramble it with the index sequence Y obtained by Lorenz mapping to obtain a new matrix I_5.

Step 6: use formula (2) to generate the DNA sequence P, and then determine the number of base substitutions according to the conversion rule. Select the corresponding rules from Table 2 to replace and form the corresponding code matrix I_6.

Step 7: after replacement, select a DNA encoding rule. Then, the binary code is formed by the replacement process, the gray value is formed by the conversion process, and the matrix I_7 is formed by the restoration process. The corresponding replacement expression is as follows:

$$\begin{cases} x_i = x_i, & P_i = 0, \\ x_i = L(x_i), & P_i = 1, \\ x_i = L(L(x_i)), & P_i = 2, \\ x_i = L(L(L(x_i))), & P_i = 3. \end{cases} \quad (7)$$

Step 8: determine the index sequence Z according to the Lorenz mapping equation (1), scramble the matrix I_7, obtain the encrypted matrix I_8, and output the corresponding ciphertext. The decryption algorithm only needs to reverse the above steps.

This algorithm also meets the applicability requirements for color image encryption. During the processing, RGB decomposition is simply performed, and then the same operation is performed.

4. Experimental Simulation Results and Analysis

The algorithm proposed in this article is used to simulate several different images. This algorithm can be used to encrypt images of any size. Figure 3 shows the encrypted and decrypted keyframe images from the visual data monitored in the industrial network. In Figure 3, the experimental results show that after using the encryption algorithm to encrypt the plaintext image, no information about the plaintext image can be obtained from the encrypted image. Thus, our proposed image encryption algorithm can

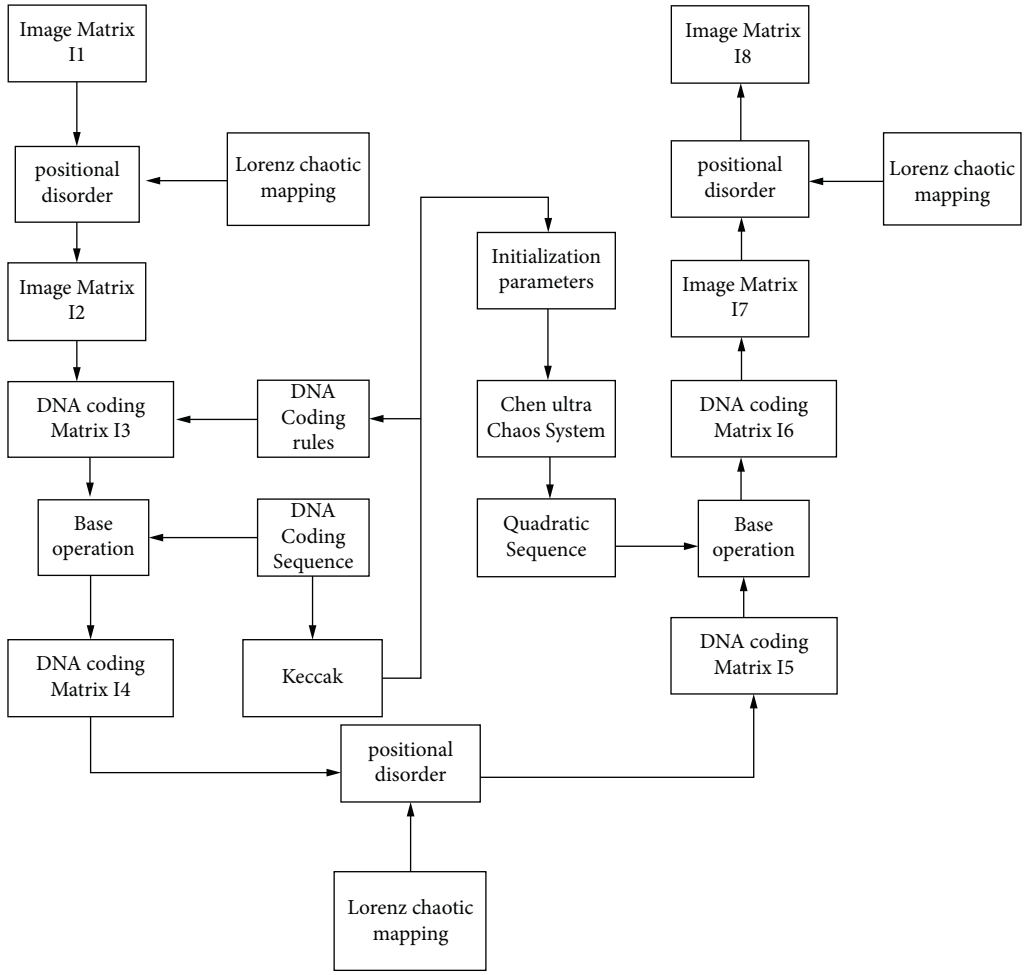

Figure 2: Encryption flowchart.

withstand statistical attacks, and the encryption algorithm has a better effect. Moreover, this algorithm is lossless, and the decrypted image is exactly the same as the original image.

4.1. Key Space and Sensitivity Analysis. The key space refers to the range of the key size used in the encryption and decryption process. The larger the value range of the key is, the larger the corresponding key space is. In theory, if the key space of the encryption system is greater than 2^{128}, it is believed that the encryption system can effectively resist exhaustive attack. In the encryption algorithm of this paper, set $(x_0, y_0, z_0, x_1, y_1, z_1,$ and $w_1)$ as the key. The accuracy of the initial value of the hyperchaotic system of the algorithm in this article can reach 10^{96}, which reaches a very high level with such a large space key. So, it can effectively resist exhaustive attack.

If a small change in the key causes the decrypted image to be completely unrelated to the original image, the key is said to be sensitive, and this property makes it possible to improve the resistance to cracking by increasing the key by a certain number of decimal places. Figure 4(d) shows the decrypted image under the wrong key obtained by changing (0), one of the keys used in encrypting the image, from 0.4985 used in encryption to just 0.4985000000000001 in decryption. It means that the key sensitivity of this algorithm is extremely strong, and even if one of the keys is changed very slightly during decryption, the decryption result is completely different from the original image without any correlation, and the number of keys of this algorithm is very large, which can completely resist the attack by exhaustive keys.

4.2. Analysis of Information Entropy. Information entropy can be used to describe the uncertainty of objects. Its calculation formula is as follows:

$$H(m) = -\sum_{k=0}^{2^N-1} p(m_i)\log_2 p(m_i). \tag{8}$$

A specific analysis shows that for grayscale images, the total number of states of information m is 256. When the information entropy is 8, the information is completely random. According to Table 6, the corresponding information entropy determined by encrypting the test images with the algorithm of this article are close to 8, and the information leakage of the image is very little. Therefore, the image encryption algorithm proposed in this paper generates secure encrypted images with randomness.

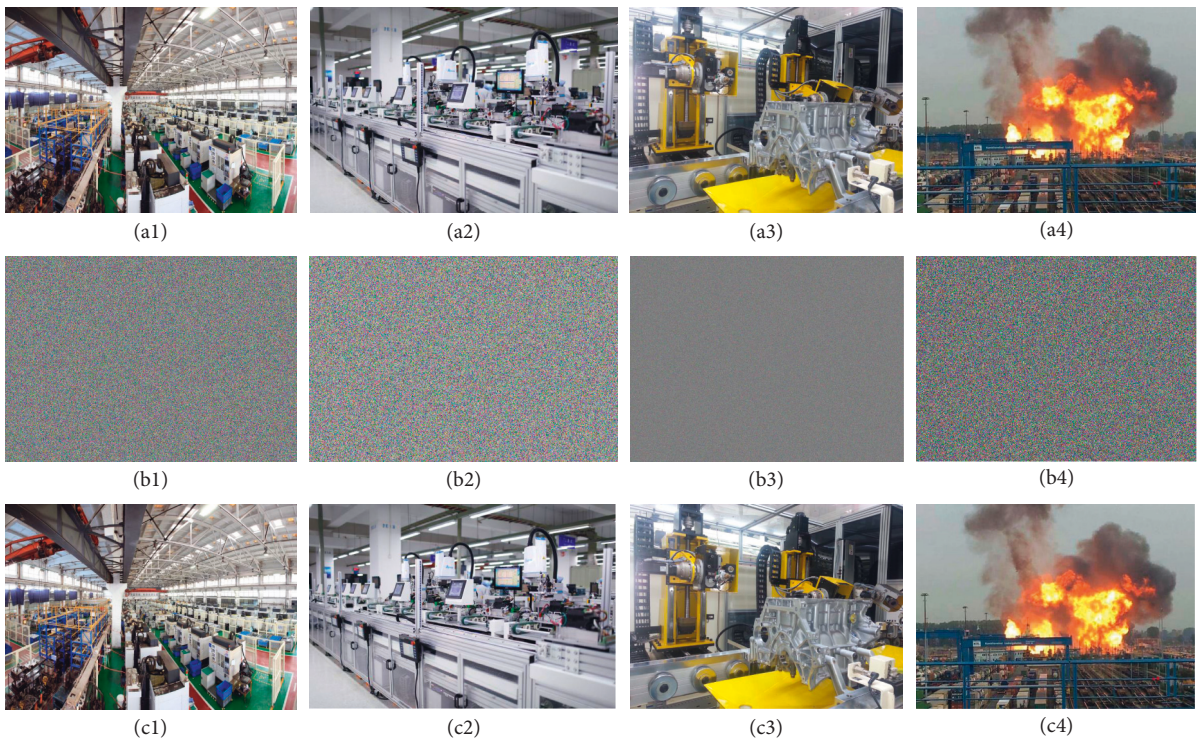

Figure 3: The results of plain, cipher, and decrypted images. (a.i) The plain images, (b.i) the encrypted images, and (c.i) the decrypted images (from left to right, $i \in \{1, 2, 3, 4\}$).

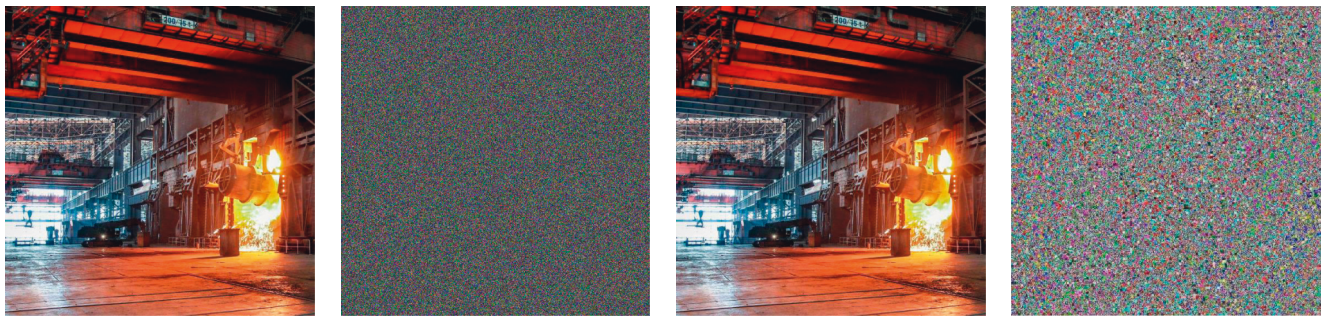

Figure 4: Decrypt image in case of key error. (a) Plain images, (b) encrypted images using the secret key, (c) the decrypted images using the secret key, and (d) small change in key.

Table 6: Information entropy tests.

Name	Plain image			Cipher image		
	R	G	B	R	G	B
Image 1	7.9216	7.9191	7.9257	7.9997	7.9997	7.9998
Image 2	7.7586	7.744	7.749	7.9993	7.9993	7.9993
Image 3	7.8626	7.8017	7.8177	8	8	8
Image 4	7.5443	7.455	7.3033	7.9992	7.9992	7.9991
Image 5	7.9538	7.5324	7.2105	7.9997	7.9998	7.9997

4.3. Analysis of Histogram. An image histogram shows the distribution of the pixel intensity values, and it provides some statistical information of the image. A secure image encryption system can make the encrypted image have a uniform histogram to resist any statistical attacks.

As shown in Figure 5, we can see that the histograms of all three channels of the original image are undulating, while the histograms of all three channels of the ciphertext image are flatly distributed with pseudorandomness, which can hide the statistical properties of the original image, and thus can effectively resist large-scale histogram-based statistical attacks against the image.

4.4. Correlation Analysis. The closer the values between adjacent pixels of an image are, the higher the correlation between adjacent pixels is. The plaintext images have high information redundancy and high correlation of neighboring pixels. In general, the original image has high correlation close to 1. Therefore, image encryption should be

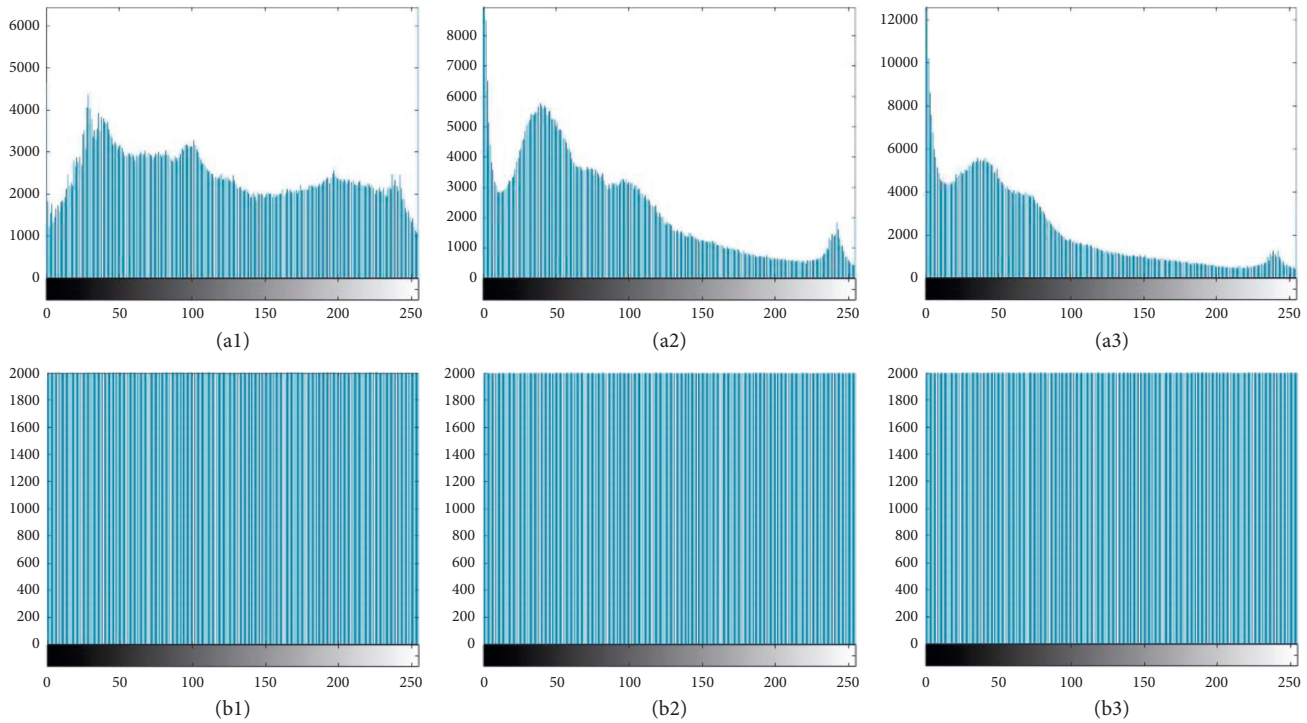

Figure 5: The histogram of the plain and encrypted images. (a.i) The R, G, and B histograms of the plain images, respectively. (b.i) The R, G, and B histograms of the encrypted images, respectively (from left to right, $i \in \{1, 2, 3\}$).

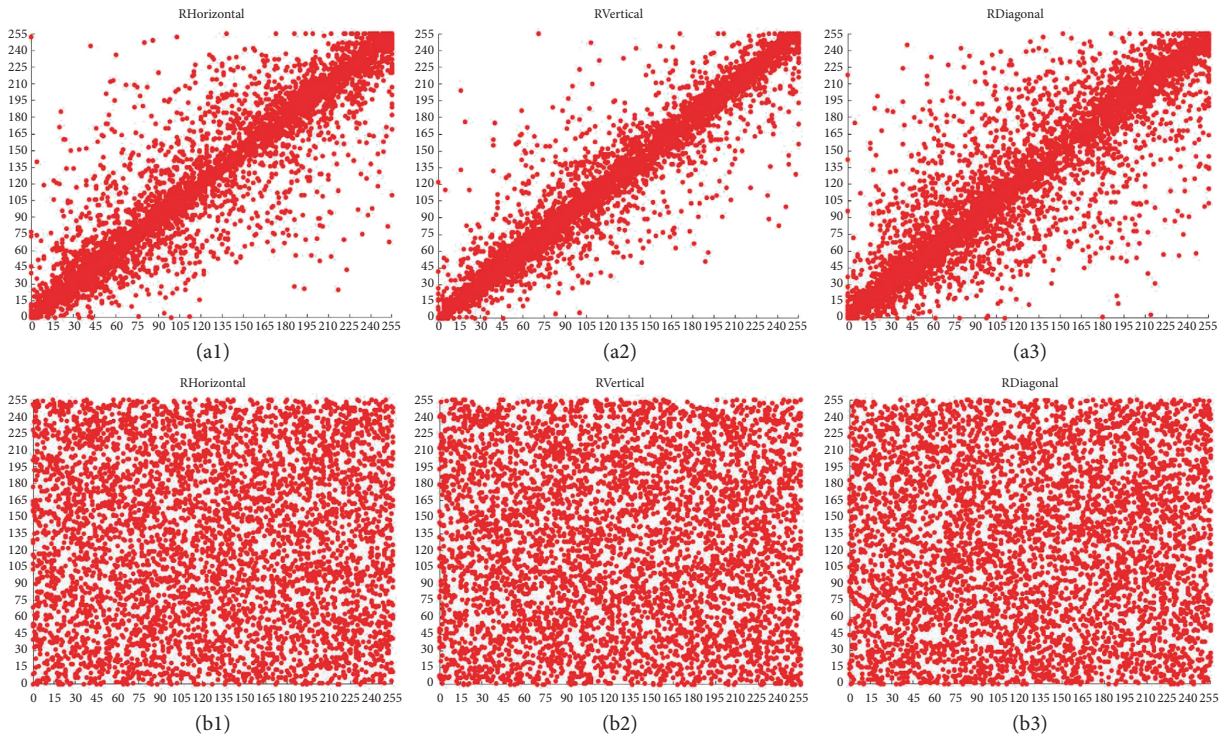

Figure 6: Correlation distribution of adjacent pixels in the plain and encrypted images in the red channel. (a.i) The horizontal, vertical, and diagonal directions of the plain images, respectively. (b.i) The horizontal, vertical, and diagonal directions of the encrypted images, respectively (from left to right, $i \in \{1, 2, 3\}$).

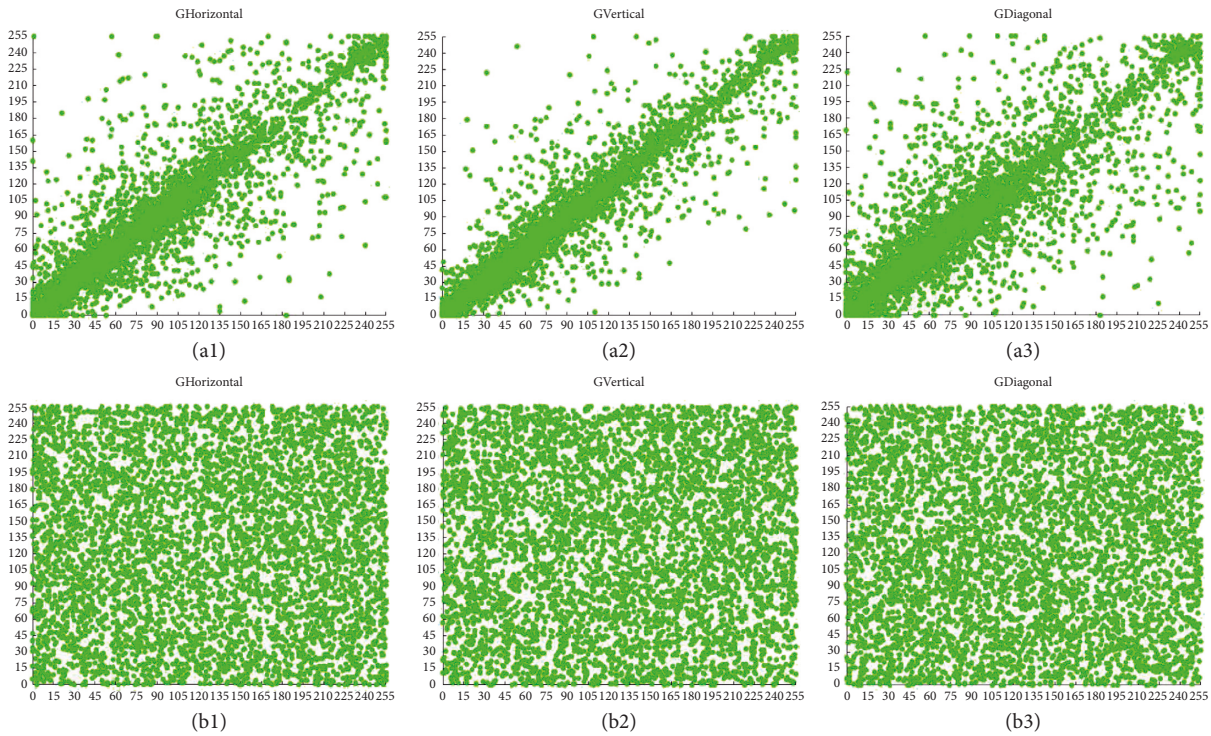

FIGURE 7: Correlation distribution of adjacent pixels in the plain and encrypted images in the green channel. (a.i) The horizontal, vertical, and diagonal directions of the plain images, respectively. (b.i) The horizontal, vertical, and diagonal directions of the encrypted images respectively (from left to right, $i \in \{1, 2, 3\}$).

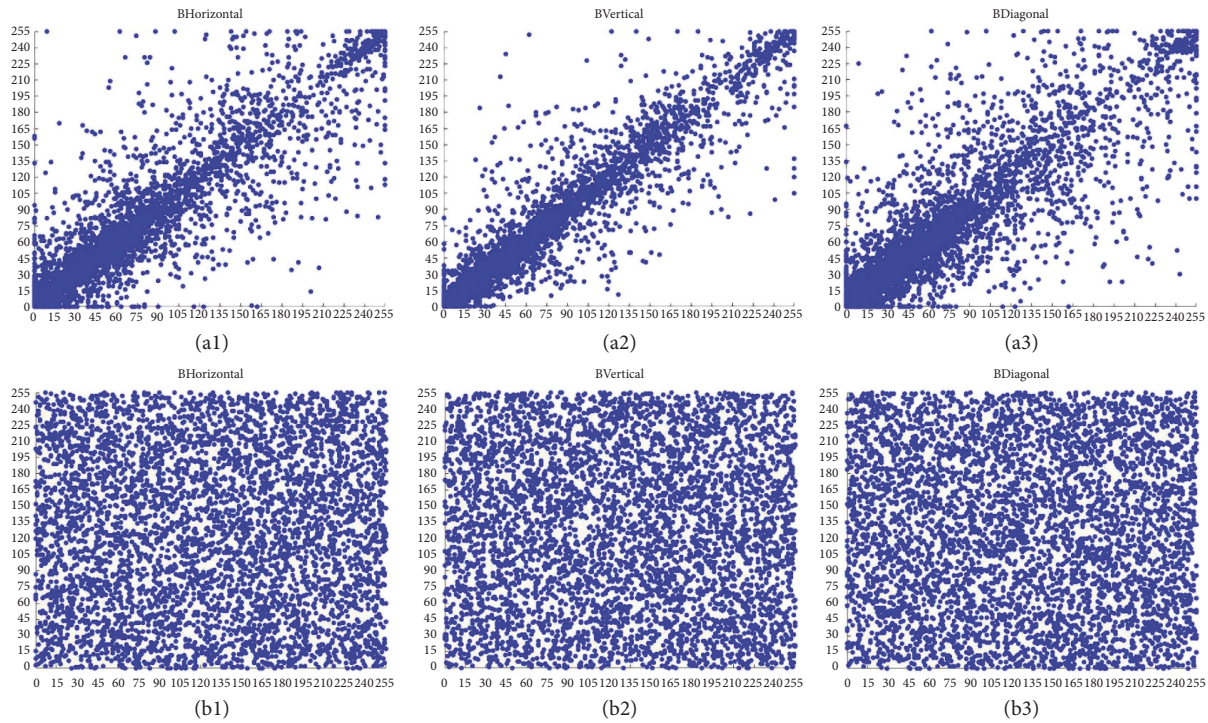

FIGURE 8: Correlation distribution of adjacent pixels in the plain and encrypted images in the blue channel. (a.i) The horizontal, vertical, and diagonal directions of the plain images, respectively. (b.i) The horizontal, vertical, and diagonal directions of the encrypted images, respectively (from left to right, $i \in \{1, 2, 3\}$).

Table 7: Relation of comparison of adjacent pixels.

	Component	Plain image			Cipher image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Image 1	R	0.94851	0.94510	0.90871	0.00193	0.00912	0.01150
	G	0.94898	0.94382	0.90848	−0.01287	−0.01729	−0.00829
	B	0.94954	0.94472	0.91027	0.00068	0.01071	0.01067
Image 2	R	0.95757	0.9382	0.90709	−0.00121	−0.01014	−0.00361
	G	0.95744	0.93797	0.90702	−0.00402	0.00738	0.003608
	B	0.95891	0.94027	0.9108	−0.00642	−0.00624	−0.00439
Image 3	R	0.99273	0.99452	0.98797	0.00875	−0.01873	−0.00453
	G	0.99219	0.99411	0.98713	0.00603	0.00229	0.00609
	B	0.99377	0.99542	0.98713	−0.0107	−0.01102	0.00624
Image 4	R	0.98484	0.96936	0.95865	0.00531	0.017608	0.02200
	G	0.97993	0.96087	0.94654	0.014381	0.00517	−0.03671
	B	0.9758	0.9494	0.93226	0.00894	−0.00344	−0.00179
Image 5	R	0.95749	0.92225	0.90197	0.11934	−0.00692	−0.01505
	G	0.94522	0.91011	0.88341	0.01001	−0.01374	−0.01734
	B	0.94662	0.91398	0.88722	−0.01810	0.00256	−0.01048

able to eliminate these correlations, and the ideal value of correlation for encrypted images should be 0 [29]. The mathematical expressions for the correlation calculation of adjacent pixels (r_{xy}) are shown as follows:

$$\begin{aligned} r_{xy} &= \frac{cov(x, y)}{\sqrt{D(x)D(y)}}, \\ cov(x, y) &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)\left(y_i - E(y)\right), \\ E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)^2, \end{aligned} \tag{9}$$

where x and y are the data values at adjacent positions, N is the log 5000 of the taken pixel points, $E(x)$ is the mean of the taken pixels, $D(x)$ is the variance, $cov(x, y)$ denotes the correlation function, and r_{xy} is the correlation coefficient. And the larger its absolute value, the stronger the correlation.

In order to resist statistical analysis attacks, it is necessary to break the strong correlation between pixels. We use a statistical test of the correlation of two adjacent pixels in an encrypted keyframe industrial image. We randomly select 5000 pixels in the keyframe. The correlation of the corresponding adjacent pixels in each channel of the RGB space of the color image are tested in the horizontal, vertical, and diagonal directions. Figures 6–8 give the visual results of the correlation distribution of two adjacent pixels in the horizontal, vertical, and diagonal directions of the keyframe image as well as the corresponding encrypted image frame image. The first row is the original image, while the second row is the encrypted image. It can be noticed that the correlations of the original image and the encrypted image are very different. The points in the plot of the correlation of the encrypted image have a good uniform probability distribution while those in the original image are concentrated on the diagonal line in the plot.

In this paper, correlations are calculated in three channels of industrial images along horizontal, vertical, and diagonal directions. Table 7 shows the correlation results of this experiment with different original images and their encrypted images. The statistical results show that the pixel correlation of the original image is very strong, while the correlation coefficient between adjacent pixels of the ciphertext image is close to zero. The algorithm in this article disrupts the correlation between pixels and resists statistical analysis attacks.

5. Conclusion

Video surveillance networks in industrial environments are growing rapidly due to the complementary role of the IOT, but at the same time, a large amount of redundant video data is being generated. This makes the transmission, analysis, and management of images difficult and challenging. This article proposes a four-dimensional hyperchaos and DNA genetics calculation to improve the image encryption method of the chaotic system. This method uses the hash value K generated by the Keccak algorithm as the initial value of the hyperchaotic system, so that the pseudorandom chaotic sequence generated by it can scramble the position of the pixel. Then, make the pixels change dynamically with the DNA code, so that the algorithm has better confusion and diffusion characteristics. Finally, a security analysis is carried out with simulation experiments, which further confirms that the image encryption algorithm proposed in this article is highly secure and can resist various types of attacks.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Key Technology Research Project of Complex Multi-User Wireless Routing Management System (no. 2018XJZD005), basic scientific research ability enhancement project for Young and Middle-Aged Teachers in Guangxi Universities (no. 2021KY0621), and basic scientific research ability enhancement project for Young and Middle-Aged Teachers in Guangxi Universities (no. 2022KY0605).

References

[1] N. Magaia, R. Fonseca, K. Muhammad, A. H. F. N. Segundo, A. V. Lira Neto, and V. H. C. de Albuquerque, "Industrial internet-of-things security enhanced with deep learning approaches for smart cities," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6393–6405, 2021.

[2] Z. Lv, L. Qiao, J. Li, and H. Song, "Deep-learning-enabled security issues in the internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9531–9538, 2021.

[3] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de-Artaza, and L. Patrono, "Internet of Things (IoT): opportunities, issues and challenges towards a smart and sustainable future," *Journal of Cleaner Production*, vol. 274, p. 122877, 2020.

[4] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent advances in the internet-of-medical-things (IoMT) systems security," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8707–8718, 2021.

[5] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, "Challenges and opportunities in securing the industrial internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 2985–2996, 2021.

[6] H. Tran-Dang, N. Krommenacker, P. Charpentier, and D.-S. Kim, "Toward the internet of Things for physical internet: perspectives and challenges," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4711–4736, 2020.

[7] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[8] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[9] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[10] M. Sajjad, I. Mehmood, and S. Baik, "Sparse representations-based super-resolution of key-frames extracted from frames-sequences generated by a visual sensor network," *Sensors*, vol. 14, no. 2, pp. 3652–3674, 2014.

[11] I. Mehmood, M. Sajjad, W. Ejaz, and S. W. Baik, "Saliency-directed prioritization of visual data in wireless surveillance networks," *Information Fusion*, vol. 24, pp. 16–30, 2015.

[12] C. Li, Y. Zhang, and E. Y. Xie, "When an attacker meets a cipher-image in 2018: a year in review," *Journal of Information Security and Applications*, vol. 48, p. 102361, 2019.

[13] K. J. Persohn and R. J. Povinelli, "Analyzing logistic map pseudorandom number generators for periodicity induced by finite precision floating-point representation," *Chaos, Solitons & Fractals*, vol. 45, no. 3, pp. 238–245, 2012.

[14] D. Yoshioka and K. Kawano, "Periodic properties of Chebyshev polynomial sequences over the residue ring $\mathbb{Z}/2{k}\mathbb{Z}$," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 8, pp. 778–782, 2016.

[15] C. Li, B. Feng, S. Li, J. Kurths, and G. Chen, "Dynamic analysis of digital chaotic maps via state-mapping networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 6, pp. 2322–2335, 2019.

[16] 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.

[17] X. Chai, Z. Gan, K. Yuan, Y. Chen, and X. Liu, "A novel image encryption scheme based on DNA sequence operations and chaotic systems," *Neural Computing & Applications*, vol. 31, no. 1, pp. 219–237, 2017.

[18] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Optics and Lasers in Engineering*, vol. 88, pp. 197–213, 2017.

[19] H. Wen, S. Yu, and J. Lü, "Breaking an image encryption algorithm based on DNA encoding and spatiotemporal chaos," *Entropy*, vol. 21, no. 3, p. 246, 2019.

[20] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[21] O. E. Rössler, "An equation for hyper chaos," *Physics Letters A*, vol. 71, no. 2-3, pp. 155–157, 1979.

[22] C.-X. Zhang, S.-M. Yu, and Y. Zhang, "Design and realization of multi-wing chaotic attractors via switching control," *International Journal of Modern Physics B*, vol. 25, no. 16, pp. 2183–2194, 2011.

[23] W. J. Ruan and Q. G. Yang, "Research on complex dynamics of a new four-dimensional hyperchaotic system with finite and infinite isolated singularities," *Journal of Guangxi Normal University*, vol. 03, 2021.

[24] J. X. Zhao and X. F. Zhang, "Spatiotemporal color image encryption method based on combined chaotic systems," *Computer Engineering and Design*, vol. 37, no. 9, 2016.

[25] Y. G. Huang, Y. X. Du, and W. Shi, "Image encryption algorithm based on a novel combinatorial chaotic mapping," *Microelectronics & Computer*, vol. 36, no. 5, pp. 47–52, 2019.

[26] G. Bertoni and J. Daement, "The Keccak sponge function family [EB/OL]," 2010, https://Keccak.noekeon.org.

[27] P. Morawiecki, J. Pieprzykand, and M. Srebrny, "Rotational crypt analysis of round-reduced Keccak [EB/OL]," 2018, https://eprint.iacr.org.

[28] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. Wang, and S. W. Baik, "Secure surveillance framework for IoT systems using probabilistic image encryption," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3679–3689, 2018.

[29] B. Norouzi, S. Mirzakuchaki, S. M. Seyedzadeh, and M. R. Mosavi, "A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1469–1497, 2014.