Electronic Research Archive

ERA, 33(8): 4933–4963.
DOI: 10.3934/era.2025222
Received: 15 July 2025
Revised: 13 August 2025
Accepted: 19 August 2025
Published: 25 August 2025

https://www.aimspress.com/journal/ERA

Research article

Hybrid chaotic image encryption in spatial-frequency domain integrated with bit-level dynamic diffusion

Yizhe Lu[1], Jianhua Song[1,2,*], Xinrong Fu[2] and Xinying Huang[2]

[1] Key Laboratory of Light Field Manipulation and System Integration Applications in Fujian Province, Minnan Normal University, Zhangzhou 363000, China
[2] College of Physics and Information Engineering, Minnan Normal University, Zhangzhou 363000, China

* **Correspondence:** Email: songjianhua@mnnu.edu.cn.

Abstract: We aimed to address the issues of low encryption complexity and insufficient resistance to statistical attacks in chaotic image encryption algorithms. This paper proposes a novel method that combines hash scrambling, and bit-level transformation, leveraging the synergy between chaotic mapping and spatial-frequency transformation. Firstly, a specific hash function is constructed using the cubic chaotic system to achieve efficient scrambling of the image. Subsequently, the frequency domain data is generated by the Fourier transform and represented in the form of an optimized complementary code. Finally, the encryption is completed by DNA primary diffusion, secondary diffusion based on adjacent blocks, and the neighbor bit deprivation operation, and the entire process is jointly regulated by the Chen chaotic system and the Sin-Tent-Cos chaotic system. The experiment demonstrates that this scheme exhibits excellent robustness and security, with all indicators surpassing those of traditional methods. In particular, its resistance to differential attacks is close to the theoretical optimal value, demonstrating the important application potential of spatial-frequency joint encryption.

Keywords: image encryption; chaotic system; hash scrambling; DRPE (double random-phase encoding); bit-level transformation; security analysis

1. Introduction

With the rapid development of communication technology, information security is receiving increasing attention. Digital images contain a large amount of private information, and the challenges

of secure transmission over public channels and secure storage within cloud environments must be addressed with urgency. Image encryption is one of the methods in cryptography which ensures images' security [1]. To deal with the problems above, the image is converted to an unrecognizable form by an encryption algorithm before transmission, and the key is transmitted together with it, which is the only way to restore the original image. Image encryption technology has a wide range of applications, including safeguarding patients' privacy in the medical industry, preserving state secrets in the military, and protecting users' privacy when developing App programs. It proves to be highly beneficial in both academic research and industrial applications.

In recent years, researchers have continuously improved the security and efficiency of encryption algorithms by integrating multidisciplinary techniques. For example, the combination of a fractional-order Hopfield neural network and differential encryption significantly improves the privacy protection performance of medical images by augmenting the key space with nonlinear dynamics [2]. Hybrid domain encryption with a watermarking technique achieved a balance between robustness and invisibility in social image security [3]. Robust quadratic polynomial hyperchaotic mapping with a pixel fusion strategy improved the efficiency of encryption by optimizing the randomness of chaotic sequences [4]. Dynamic vector-level operations with two-dimensional (2D)-enhanced logistic modular maps have shown improved efficiency in chaotic image encryption [5]. These works provide an important reference for this research into the optimization of chaotic systems and the integration of multiple techniques.

A chaotic system is a kind of complex dynamic system, which has the characteristics of pseudo-randomness, initial sensitivity and traversability, and is widely used because it meets the requirements for image encryption sequences [6]. However, chaotic systems still suffer from issues such as insufficient key sensitivity and long-term predictability, for which researchers are trying to introduce other techniques to improve encryption security. Some have considered adding quantum random walks to enhance cryptosystems [7], combined with a convolutional self-encoder to increase the diffusion effect and achieve fusion with deep learning [8]. Integration with hashing algorithms has become a significant research focus in recent years, with examples including hash-driven chaotic key creation [9] and hash-enhanced diffusion mechanisms [10,11].

DNA computation in genetics has advantages such as high parallelism, and many scholars have tried to apply it to the field of image encryption. For example, generating arithmetic rules with chaotic mapping [12–14] can significantly expand the key space and enhance the resistance to statistical attacks. Nematzadeh et al. proposed a new image encryption method by combining the properties of DNA sequences and binary search trees [15]. Zhang et al. proposed an image encryption algorithm based on dynamic DNA encoding [16]. Bit-level arithmetic has also received attention from researchers due to its good properties, such as large storage and low energy consumption. It directly manipulates the bit-plane of pixels to enhance the security and complexity of encryption through scrambling, diffusion, obfuscation, etc. In recent years, it has gradually evolved into a profound fusion of multiple techniques, such as using chaotic mapping to regulate bit operations [17] and combining encryption algorithms with bit-level diffusion [18]. There are also encryption approaches that use quantum DNA encoders [19] to produce random sequences and perform quantum bit substitution at the bit level [20]. Some studies [21] have implemented bit-level transformations and have controlled global scrambling via feedback, leading to quick and secure encryption.

The image processing technique of fused optical transformation is a current research hotspot, which is preferred by researchers due to its advantages of multidimensional information storage, vast

capacity, and high-speed parallel processing capability. For example, combining optical chaos [22], compressed perception [23], and deep learning [24] in the frequency domain suggests a number of security-enhancing approaches. Furthermore, multimodal fusion employs unique optical techniques such as polarization encryption [25], metasurface encryption [26], and others to alter the optical field, thus improving security. Zhang et al. provide an image encryption technique that is resistant to quantum assaults by using quantum measurements [27].

The research on the security of cryptanalysis reveals the key vulnerabilities of existing image encryption schemes and provides an important reference for designing more robust algorithms. The security analysis of medical image encryption based on chaos and DNA [28] proves its sensitivity to selective plaintext attack. Cryptanalysis based on 2D hyperchaotic mapping [29] emphasizes the necessity of enhancing nonlinear operations. These studies show that an excellent encryption scheme must be able to resist specific attacks such as differential cryptanalysis and chosen ciphertext attack. To this end, this study employs hash key generation and transforms image information into the frequency domain for secondary encryption to ensure that this scheme meets the security requirements of modern cryptanalysis. The goal is to enhance the encryption complexity and increase resistance to statistical attacks. The contributions of this paper are as follows:

(i) We construct a scrambling framework that introduces an optimized hash table. Compared with traditional methods, it shows significant advantages in terms of the initial obfuscation effect.

(ii) A hybrid diffusion strategy coupled with an optimized complementary representation and bit-level algorithm is proposed to enhance the encryption complexity and security.

(iii) Optimization of the classical diffusion method, combined with hash scrambling to diffuse information separately in the spatial-frequency domain, considerably increases the encryption system's robustness.

(iv) The proposed algorithm is not only close to the theoretical optimum in the test results for differential attacks, but also shows superior robustness, which makes it suitable for application scenes with high security requirements.

The remaining parts of this paper are organized as follows: Section 2 introduces the related theory, Section 3 describes the innovative scheme, Section 4 presents the encryption and decryption processes in detail, Section 5 analyzes the security of this algorithm, and Section 6 concludes the paper.

2. Related theory

2.1. Chaotic systems

Chaotic phenomena are prevalent in deterministic nonlinear systems, which exhibit stochastic behavior and extreme sensitivity to minor changes in the initial conditions and the system's parameters [30].

2.1.1. Cubic chaotic systems

Cubic chaotic mapping [31] has better chaotic traversal as described in Eq (1).

$$x(i+1) = ax(i)\left[1 - x(i)^2\right], \tag{1}$$

where a is the mapping factor. The cubic chaotic mapping value x has better traversal in the interval

(–1, 1), when a is in the interval (2.5, 3). The chaotic mapping diagram is shown in Figure 1.

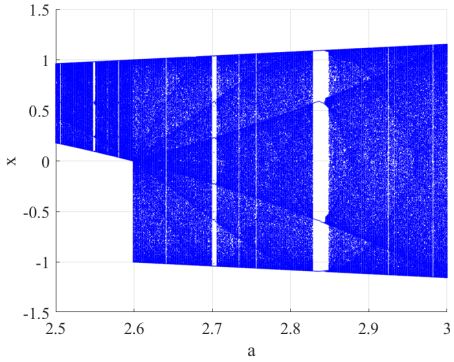


Figure 1. Cubic chaotic map.

2.1.2. The Sin-Tent-Cos chaotic system

The Sin-Tent-Cos chaotic system [32] is a mixture of mappings by considering sine mapping, cosine mapping, and tent mapping as seed mappings via Eq (2)

$$\begin{cases} x(i+1) = \cos\left[\pi(r\sin(\pi \cdot x(i)) + 2(1-r)x(i) - 0.5)\right], & x_i < 0.5 \\ x(i+1) = \cos\left[\pi(r\sin(\pi \cdot x(i)) + 2(1-r)(1-x(i)) - 0.5)\right], & x_i \geq 0.5 \end{cases}. \tag{2}$$

where r is a control parameter and the system is in a chaotic state when $r \in (0.01,1)$, as shown in Figure 2.

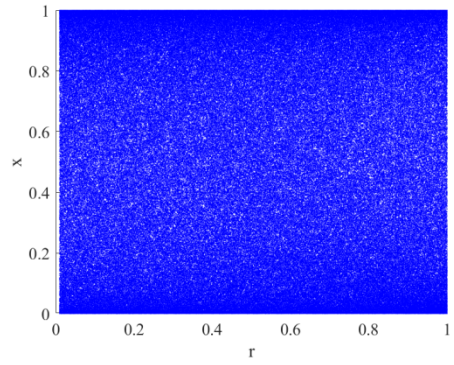


Figure 2. Sin-Tent-Cos chaotic map.

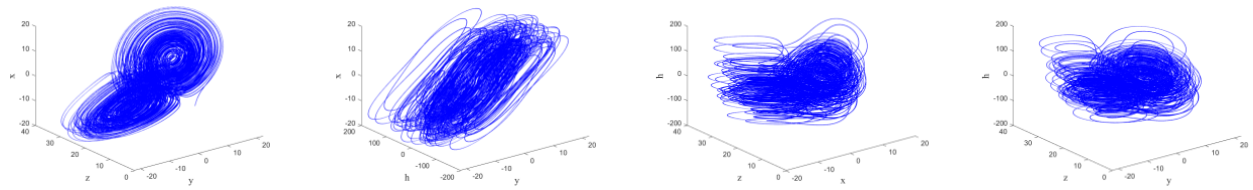
2.1.3. Chen hyperchaotic system

Compared with one-dimensional (1D) chaotic systems, high-dimensional systems process more control parameters and exhibit more complex dynamic features, as well as sensitivity to the initial conditions. Combining them with image encryption algorithms can effectively extend the key space. Therefore, we employ the Chen hyperchaotic system [33], the dynamic equations of which are defined

as shown in Eq (3)

$$\begin{cases} dx/dt = a \cdot (y - x) + h \\ dy/dt = p \cdot x + c \cdot y - x \cdot z \\ dz/dt = x \cdot y - b \cdot z \\ dh/dt = y \cdot z + q \cdot h \end{cases} \tag{3}$$

where dx/dt is the derivative of the system variable x with respect to time t; a, b, c, p, and q are the parameters of the system, which is chaotic when $a = 35$, $b = 3$, $c = 12$, $p = 7$, and $q = 0.5$, respectively. The Matlab built-in ode45 function is used to calculate the dynamic equations of Eq (3), and the solution yields four 1D chaotic sequences. Figure 3 shows the four chaotic attractor phase diagrams of *y-z-x*，*y-h-x*，*x-z-h*，and *y-z-h* plotted from the data.



(a) (b) (c) (d)

Figure 3. Attractor phase diagram of the Chen hyperchaotic systsem.

2.1.4. Basis for system selection

In this paper, the combination of three chaotic systems is chosen on the basis of the following considerations: (i) The 1D structure of the cubic chaotic system is iterative and fast, and its generated sequences still exhibit good pseudo-randomness with finite accuracy, which makes it suitable for efficiently realizing pixel-level substitution. (ii) The Sin-Tent-Cos chaotic system enhances the unpredictability of the trajectories by mixing sinusoidal, tent, and cosine mappings; realizes intermediate-level diffusion through nonlinear coupling; and has the property of efficiently resisting phase-space reconstruction attacks. (iii) The Chen hyperchaotic system has four-dimensional dynamics and possesses two positive Lyapunov exponents, which significantly improves the key space and sensitivity to the initial conditions, providing a stronger stochastic basis for the diffusion phase. When the three act in concert, the cubic mapping is responsible for fast initial scrambling to break spatial correlations, the Sin-Tent-Cos system breaks periodicity through perturbation to realize nonlinear intermediate diffusion, and the Chen system ensures that the statistical properties of the final ciphertext are close to the ideal random distribution. This layered design balances efficiency and security.

2.1.5. Analysis of pre-iteration

The initial iteration of the chaotic system is run 3000 times and discarded, mainly on the basis of two needs. (i) Elimination of transient effects: The chaotic system may fall into a short period of non-

chaotic transient behaviors, such as fixed points or periodic oscillations, due to the limitation of computational precision in the initial stage. Pre-iteration ensures that the system enters a stable chaotic state. (ii) Enhanced randomness: pre-iteration reduces the correlation between the initial key and the generated sequence, and prevents the attacker from deducing the key in reverse by choosing a plaintext attack, for example. Experiments show that the sequence entropy value of Chen's system converges to the theoretical maximum when the number of pre-iterations is >3000, which verifies the reasonableness of the operation.

2.2. Double random-phase encoding

At present, the image encryption technique combined with optical transformation has attracted extensive attention from researchers. Double random-phase encoding (DRPE) is one of the most critical techniques in this field, and the basic principle is shown in Figure 4.

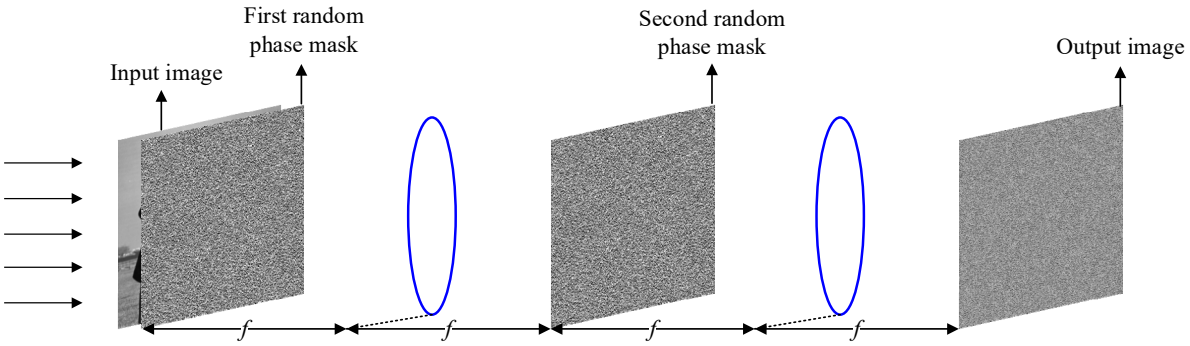


Figure 4. DRPE processing.

Two mutually independent random phase masks are placed on the input plane and the focal plane of the lens of the optical 4*f* system. The input image passes through the first lens after modulation by the first phase mask plate, and then passes through the second lens after modulation by the second phase template, which ultimately encrypts the image located on the input plane as a complex smooth white noise.

The random phase masks M_1 and M_2 are generated using chaotic mapping and their values are distributed in the range between [0,1]. The DRPE processing can be described as shown in Eq (4)

$$E(\xi,\eta) = IFT\left(FT\left(I(x,y)\cdot e^{i2\pi n(x,y)}\right)\cdot e^{i2\pi b(\mu,\upsilon)}\right), \tag{4}$$

where (ξ, η) are the coordinates of the acceptance plane; $FT(\cdot)$ and $IFT(\cdot)$ denote the Fourier transform and the inverse Fourier transform, respectively; $e^{i2\pi n(x,y)}$, $e^{i2\pi b(\mu,v)}$ are phase masks located in the image plane and the Fourier plane; i denotes the imaginary unit; and E is the obtained complex numerical data. Similarly, the decoding process is the inverse process of DRPE, which is mathematically described in Eq (5)

$$D(x,y) = \left|IFT\left(FT\left(E(\xi,\eta)\cdot e^{-i2\pi b(\mu,\upsilon)}\right)\right)\right|, \tag{5}$$

where D is the decoded image, $e^{-i2\pi b(\mu,v)}$ is the complex conjugate matrix of $e^{-i2\pi b(\mu,v)}$, and $|\cdot|$ denotes the modulus operation. Since the first phase mask M_1 is removed by the modulo operation in Eq (5), it can be omitted in the image decoding process.

2.3. DNA calculations

The deoxyribonucleic acid (DNA) sequence consists of four different nucleotide bases: Adenine (A), thymine (T), cytosine (C), and guanine (G). If represented using binary coding, the total number of possible coding combinations is 4! = 24. However, due to the bases' complementary pairing rules, i.e., A pairs with T and C pairs with G, which corresponds to the pairing of 0s and 1s in binary, then there are only eight combinatorial rules left for the DNA sequence [34,35]. Table 1 shows these eight rules.

Table 1. The rules of encoding DNA.

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

In this paper, dynamic DNA coding rules are adopted, and the rules are dynamically selected from the sequence values generated by the Chen hyperchaotic system to avoid the statistical feature leakage caused by fixed rules. Complementary base pairs are selected to encode data to strengthen the diffusion effect. For the same decimal number, the results under different encoding rules are different. Assuming that the pixel value of an image is 167 and the binary code is "10100111", if Rule 1 is selected, the DNA encoding result will be "GGCT". Assuming that Rule 5 is selected for decoding, the decoded binary will be "11110001", with a pixel value of 241. Similarly, different decoding rules can be used to dynamically obfuscate the encoding result. The decoding rules are shown in Table 2.

Table 2. The rules of decoding DNA.

Rules	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

In this paper, we use five binary arithmetic rules: addition, subtraction, multiplication, same-or, and different-or. For example, the binary addition operation and the same-or operation under the influence of Rule 1 in Table 1 are shown in Figure 5.

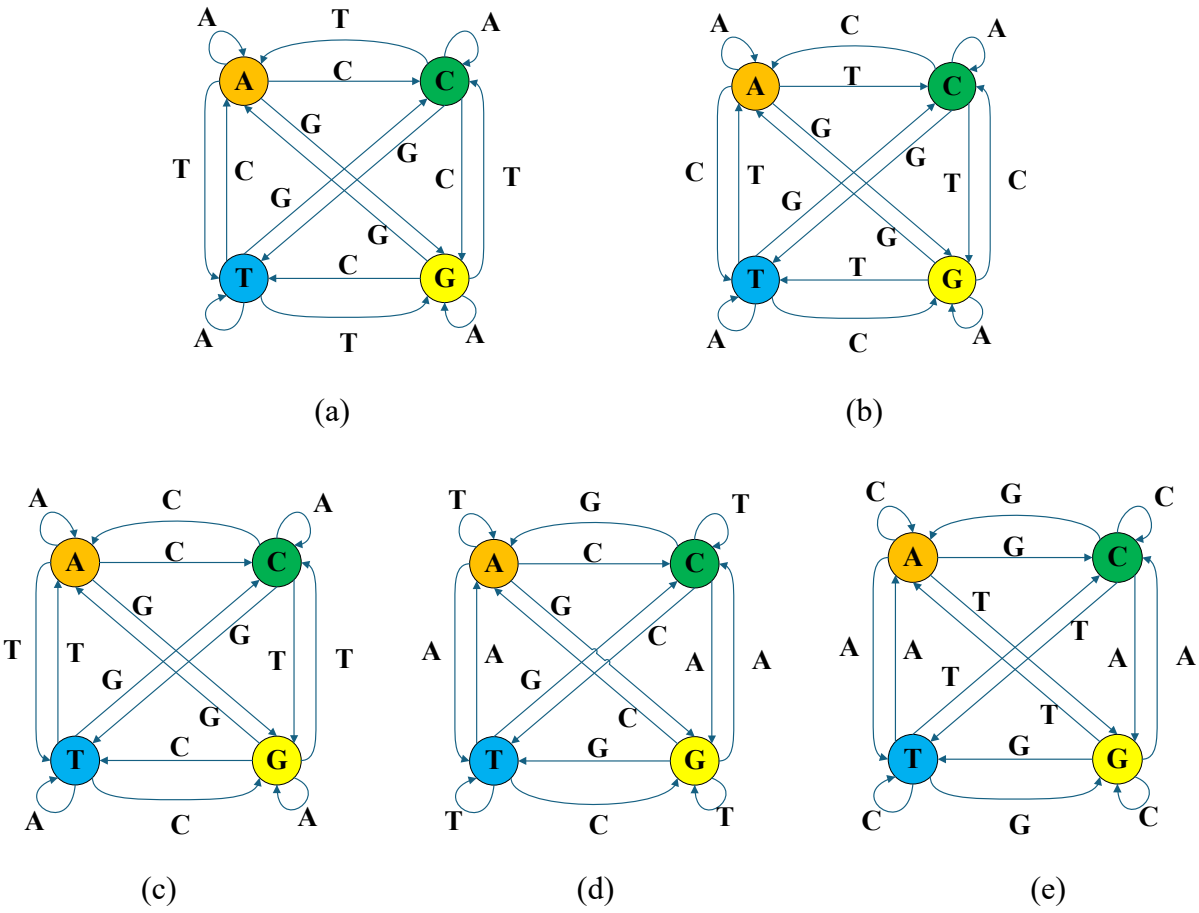


Figure 5. DNA operations. (a) Add operations, (b) Subtract operations, (c) XOR operations, (d) XNOR operations, (e) MUL operations.

3. Innovative scheme

In this section, several innovative schemes that we have devised are presented.

3.1. Pixel scrambling of hash tables

Traditional scrambling methods have low sensitivity and limited randomization, and lack data correlation. We add a hash function to improve the sensitivity, randomness, and irreversibility of keys. A hash table is a data structure that determines the position of a key on the basis of its value, allowing the result to be different from the input value and order. It is implemented through a specific function, as shown in Definition 1.

Definition 1.

$$[V,S]=\mathrm{mod}\left(\left(\mathrm{Hash}\left(Key\pm d^2\right)\right),p\right)$$

where mod() is the modulo operation and p is the length of the hash table, which is taken as a prime number of 4 * k + 3 (k is a positive integer), and the loading factor lies between 0.7 and 0.8. Moreover, the collision problem is handled using the open addressing method with d_i as an increment. For example, take p = 19 and set the keyword sequence Key to get the output result V and the sequence S

as shown in Figure 6.

0.2159	0.9501	0.8810	0.0129	0.1083	0.1040	0.2629	0.2777	0.0066	0.5678	0.5297	0.5751	0.8126	0.0364

(a) Original matrix *Key*

0.9501	0.5297	0.2777	0.8126	0.2629	0.0364	0.0066	0.2159	0.8810	0.1040	0.0129	0.5678	0.5751	0.1083

(b) After Hash scrambling *V*

2	11	8	13	7	14	9	1	3	6	4	10	12	5

(c) After sorting *S*

Figure 6. Hash scrambling of pixels.

We construct a hash-driven scrambling method that is tightly correlated with image content and has an avalanche effect on small changes in the input. The disambiguation parameters associated with the plaintext are dynamically generated. This avoids fixed-mode attacks and greatly improves the efficiency and security of encryption.

3.2. Complementary representation of floating-point numbers

The current floating-point number coding representation suffers from the problems of accuracy loss and limited dynamic range. In order to overcome these problems, we propose adaptive floating-point representation and combine the advantages of fixed-point and floating-point numbers to co-code the integer portion and high-frequency decimal portion, to reduce the cumulative error during the simulation process, and to achieve higher numerical accuracy with limited resources. In digital coding, there are three ways to represent signed integers as binary: original, inverse, and complement. For the integer –148, its three representations are shown in Figure 7(a). On this basis, we extend the complementary representation to the floating-point range, i.e., the fractional part is also represented in binary, provided that the sign and integer bits remain unchanged, as given by Definition 2.

Definition 2.

$$F = (-1)^s \times \left(\sum_{i=0}^{k-1} b_i \times 2^i + \sum_{j-1}^{m} b_{-j} \times 2^{-j} \right),$$

where s is the sign part, k is the integer part, and m is the decimal part.

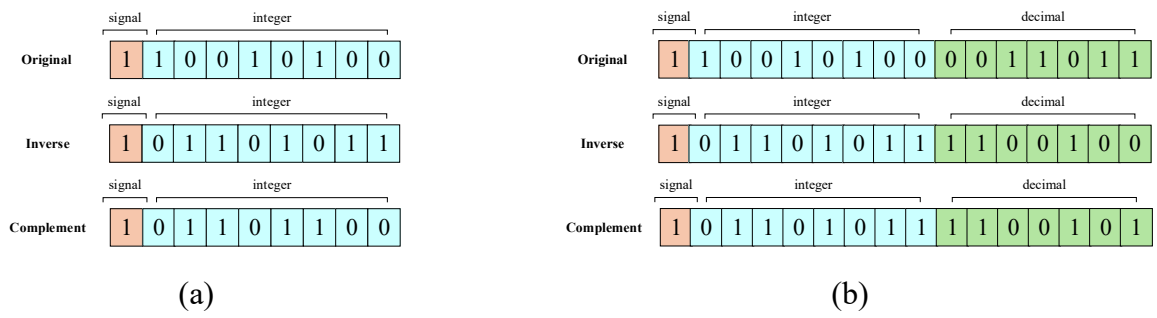


Figure 7. Binary code representation.

Assume that the floating-point number is –148.2109375. We use one bit for the sign, eight bits for the integer, and seven bits for the decimal. The binary conversion steps are as follows: the original code of the integer part is "10010100", and the original code of the decimal part is "0011011". Finally, it is combined and output in the form of complement, as shown in Figure 7(b).

The representation has significant advantages: It extends the range of complementary representation. Compared with the original code, it has a larger range of numerical representation under the same bit width, and it supports the unification of symbolic and numerical bit operations, while eliminating the sign ambiguity of zero. Therefore, this paper adopts the complementary code for data encoding.

3.3. Neighboring bit deprivation (NBD)

Due to the significant statistical characteristics of image data itself, coupled with the lack of a dynamic update mechanism in the encryption process, traditional encryption schemes tend to be less effective in the face of statistical analysis attacks.

In order to reduce the correlation and enhance the encryption effect, we design a new bit-level transformation scheme called neighbor bit deprivation. Assume that a pixel value P is 134, with the binary sequence "10000110", and the neighboring pixel's value Q in the horizontal direction is 133, with the binary sequence "10000101". Defining the first left digit as the contrast bit, according to Definition 3, since the contrast bits of P and Q are equal, the NBD principle is triggered to change the contrast bit data of the latter, and then the value of Q is changed according to Definition 4 using the data of the chaotic sequence. In this way, a pair of data points with strong correlation is changed. Next, NBD is performed for different directions. It has been proved experimentally that this scheme can effectively attenuate the correlation between neighboring pixels, improve the algorithm's resistance to statistical analysis attacks, and demonstrate excellent security in the final analysis of experimental results.

Definition 3.

$$IsE_1 = \begin{cases} 1, & \text{bitand}(P,128) = \text{bitand}(Q,128) \\ 0, & \text{bitand}(P,128) \neq \text{bitand}(Q,128) \end{cases}$$

Definition 4.

$$\begin{cases} P = P + C_h, & IsE_1 = 1 \\ P, & IsE_1 = 0 \end{cases}$$

where bitand() denotes the bitwise-and operation.

NBD improves security by destroying the local correlation between pixels. The steps are as follows: 1) Decompose the pixel matrix into 8-bit planes. 2) For the target bit B, extract all the bit values in its neighborhood. 3) Discriminate the same bit values using Definition 4. 4) If bit deprivation is triggered, the chaotic sequence is utilized for perturbation. 5) Repeat Steps 2)–4) in different directions until processing is complete. Analyzed in terms of security, the NBD algorithm extends the influence of the modification of a single bit to the neighborhood, significantly enhancing the avalanche effect. Combined with the dynamic operation selection of chaotic sequences, it can effectively resist differential attacks.

As shown in Figure 8, for a 4 × 4 pixel matrix, the first, second, and third bits are set as the contrast bits, and A, B and C are the operation matrices, which undergo three operations in the horizontal, vertical, and diagonal directions, respectively. It can be seen that the final result has a large change compared with the initial conditions.

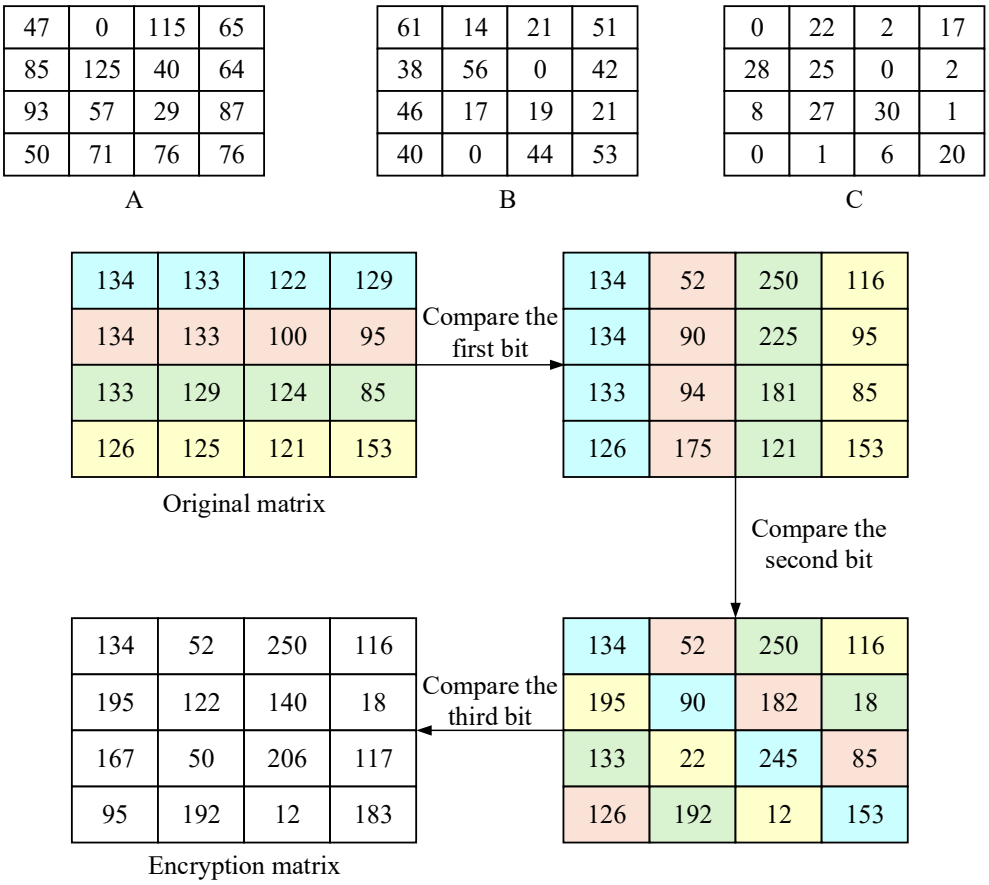


Figure 8. Neighborhood bit deprivation program.

3.4. Technology integration details

In this paper, the key parameters of the hash table are dynamically generated by a chaotic system, and the initialization of the hash parameters is driven by the Chen hyperchaotic system. The chaotic sequences are processed by specific quantization to dynamically determine the core parameters of the hash table, including the prime length as well as the incremental parameter, to ensure the mathematical completeness of the hash structure, and, at the same time, to enhance the conflict resolution capability. This design makes the hash structure strongly associated with the key, and any small changes in the key will lead to significant changes in the hash parameters through the sensitivity of the chaotic system.

For disambiguation processing, the hash value of the plaintext pixel block is first calculated and nonlinearly fused with the chaotic sequence. Dynamic disambiguation parameters are generated and used as the basis for selecting DNA encoding rules. This method effectively avoids the security risks associated with fixed mapping rules in traditional DNA encoding, and generates unique encoding rules for each pixel block through the dynamic properties of the chaotic system, which significantly improves the system's ability to resist known plaintext attacks.

The experimental results indicate that this mechanism enables the DNA coding rule space to reach the order of 2^{256}, providing an additional security dimension for the system.

4. Encryption and decryption process

4.1. Encryption algorithms

In this paper, I is the original image to be encrypted, which has dimensions of $M \times N$, where M is the width and N is the height of the image in pixels. The architectural structure of our proposed encryption process is shown in Figure 9.

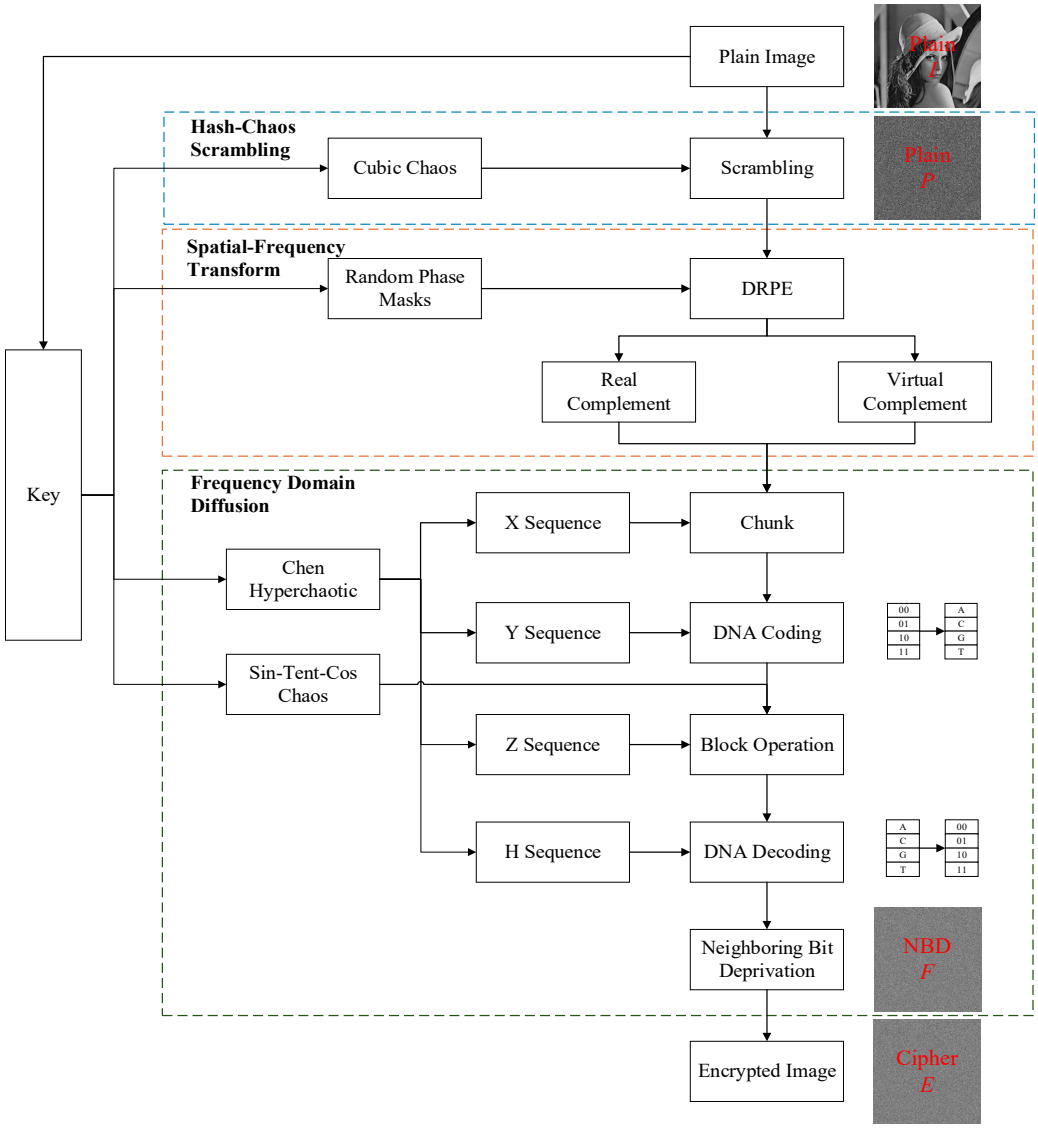


Figure 9. Encryption flowchart.

The encryption algorithm is divided into three main stages. An independent key is generated using the preprocessed original image. The first stage is hash-chaos scrambling, where a key-controlled cubic chaotic sequence is used to scramble the image pixels via a hash algorithm to obtain the image P. The second stage is the spatial-frequency transformation stage, where the image is transformed into frequency domain information using an optical 4f system and converted into a binary data stream using floating-point coding. The final stage is frequency domain processing to obtain the image F and finally

diffusion at the bit level using NBD to obtain the encrypted image E. As shown by the experimental analysis, the density of our proposed scheme shows good results in resisting the attacker's attempts.

4.2. Encryption algorithms

4.2.1. Image preprocessing and key generation

We start by standardizing the preprocessing operation of the original image. The dynamic encryption key is generated on the basis of the initial image to achieve the effect of "one encryption at a time", which lays the foundation for the subsequent steps.

Step 1: Read the image of size $M \times N$. Its gray value matrix is populated so that it satisfies Eq (6) to obtain image I, and the populated data value is 0.

$$\begin{cases} \mathrm{mod}(M,t)=0 \\ \mathrm{mod}(N,t)=0 \end{cases}, t=\lceil M/8 \rceil \tag{6}$$

where $\lceil\rceil$ is the downward rounding.

Step 2: Use the *SHA*-512 method to obtain the external key $k_1k_2\ldots k_{512}$ and use arithmetic to generate the parameter H_1 to H_{10} as shown in Eq (7).

$$\begin{cases} A_t = k_{8t-7}\cdots k_{8t-4}, & B_t = k_{8t-3}\cdots k_{8t}, & 1\le t\le 4 \\ A_t = k_{16t-47}\cdots k_{16t-40}, & B_t = k_{16t-39}\cdots k_{16t-32}, & 5\le t\le 10 \\ H_n = (A_t \oplus B_t)\oplus((A_t \,\&\, B_t)<<1)\oplus((A_t \mid B_t)>>1), & & 1\le n\le 10 \end{cases} \tag{7}$$

where $\oplus$ is the XOR operation, & is the AND operation, | is the OR operation, << is a binary left-shift by one bit, and >> is a binary right-shift by one bit.

4.2.2. Hash-chaos scrambling

The image is transformed into a scrambling sequence by the hash scrambling algorithm and then combined with a chaotic sequence for a nonlinear obfuscation operation, which destroys the spatial correlation.

Step 1: Use cubic chaos to obtain a chaotic sequence with the initial values controlled by H_1. Pre-iterate it 3000 times to obtain the chaotic sequence C_u.

Step 2: Perform an operation on image I as shown in Algorithm 1, and rearrange it to image P.

Initialize two all-zero matrices, namely T as the position mapping table and P as the output sequence. The algorithm normalizes the chaotic sequence C_u to generate an integer sequence C. In the position mapping phase, the initial position m is obtained by performing a modular L operation on each element $C(i)$. If the target position $T[m+1]$ is not occupied, the current index i is recorded and marked as occupied. In the case of conflict, the new candidate position s is calculated by using the quadratic probing method until an available position is found. Finally, by traversing the mapping table T, the original indexes corresponding to all occupied positions are sequentially stored in the output sequence P.

Algorithm 1： HT_Perm

Input： Order C_u, length *len*

Output： Image after scrambling *P*

```
L ← Prime (1.33 * len), Prime () take the largest prime number
T ← zeros (len, 2), P ← zeros (n), C ← abs (fix (100000 * Cu))
for i ← 1 to len do
    m ← mod (C, L)
    if T (m + 1,2) = 0 then
        T (m + 1,1) ← i, T (m + 1, 2) ← 1
    else
        s = mod (m ± j^2)
        if T (s, 2) = 0 then
            T (s, 1) ← i, T (s, 2) ← 1
        end
    end
end
for k ← 1 to L do
    if T (k, 2) ~= 0 then
        P ← T (k, 1)
    end
end
```

4.2.3. Frequency domain processing

The chaotic sequence controlled by the key is used to generate the phase mask. The image is converted to the frequency domain by DRPE, and we quantize the coefficients into binary, which prepares for the next step of frequency domain information encryption processing.

Step 1: Obtain a chaotic sequence by cubic chaos with the initial values controlled by H_2 and H_3. Pre-iterate 3it 000 times and quantize it into a binary image *B* as shown in Eq (8). Then equalize *B* to obtain two matrices U_1, U_2.

$$B(x)=\begin{cases}1, & U(x)\geq 0.5\\ 0, & U(x)<0.5\end{cases}. \tag{8}$$

P, as the input image, is processed by DRPE to obtain image *O*. The real and imaginary parts are converted into binary, and chunked to obtain the blocks O_1 to O_{128}.

Step 2: Obtain the chaotic sequence by the Chen hyperchaotic system, where each initial parameter is controlled by H_4 to H_7. Pre-iterate it 3000 times to obtain the sequences *X*, *Y*, *Z*, *H*. Combine the image blocks as shown in Eq (9).

$$R=\mathrm{mod}\left(\left(\mathrm{Hash}\left(\lfloor 100000*X\rfloor \pm d^2\right)\right),179\right) \tag{9}$$

where $\lfloor\rfloor$ is the downward rounding, which uses *R* to merge every four blocks of the image block into one block to obtain the disordered image blocks Q_1 to Q_{32}.

Step 3: Calculate Q according to Eq (10), denoted as T_1 to T_{32}.

$$Q_k = \text{bitand}\left(Q, 3 \cdot 4^{8-k}\right) / 4^{8-k}, k = 1,2,\cdots 8 \tag{10}$$

4.2.4. DNA diffusion and neighbor bit deprivation

The information is transformed into a DNA sequence, and the diffusion is enhanced by the principle of complementary pairing. The neighbor bit deprivation algorithm is used to carry out bit-level disturbance on the local neighborhood to further eliminate the statistical characteristics. Finally, the encrypted image is obtained.

Step 1: Obtain a sequence by Sin-Tent-Cos chaos with the initial values controlled by H_8 and H_9, Pre-iterate it 3000 times and process it into S, similar to the process in Eq (10).

Step 2: DNA-encode T to get the sequence D_e, following the rule shown in Eq (11). D_e is operated with S, and the nonprime block is run again with the previous block to obtain D_c, following the rule shown in Eq (12). Apply DNA decoding as shown in Eq (13) to obtain D.

$$\begin{cases} Rule_i = \text{mod}\left(\left|\text{fix}\left(10000 * Y\right)\right|, 8\right) + 1, & i = 1 \\ Rule_{i+1} = \text{mod}\left(Rule_i + 1, 8\right) + 1, & i > 1 \end{cases} \tag{11}$$

$$\begin{cases} Rule_i = \text{mod}\left(\left|\text{fix}\left(10000 * Z\right)\right|, 5\right) + 1, & i = 1 \\ Rule_{i+1} = \text{mod}\left(Rule_i + 1, 5\right) + 1, & i > 1 \end{cases} \tag{12}$$

$$\begin{cases} Rule_i = \text{mod}\left(\left|\text{fix}\left(10000 * H\right)\right|, 8\right) + 1, & i = 1 \\ Rule_{i+1} = \text{mod}\left(Rule_i + 1, 8\right) + 1, & i > 1 \end{cases} \tag{13}$$

Step 3: Pixelate D into a pixel matrix F. The formula is shown in Eq (14).

$$F_i = 64 * D_{4i-3} + 16 * D_{4i-2} + 4 * D_{4i-1} + D_{4i} \tag{14}$$

Step 4: The chaotic sequence is obtained by cubic chaos with the initial value controlled by H_{10}. Pre-iterate it 3000 times to obtain W and process it as shown in Eq (15).

$$\begin{cases} C_h = floor\left(\text{mod}\left(floor\left(100 * W\right) * 128, 128\right)\right) \\ C_v = floor\left(\text{mod}\left(floor\left(10000 * W\right) * 64, 64\right)\right) \\ C_d = floor\left(\text{mod}\left(floor\left(1000000 * W\right) * 32, 32\right)\right) \end{cases} \tag{15}$$

where C_h, C_v, and C_d represent the horizontal, vertical, and diagonal scrambling data respectively. We take the horizontal as an example, which is shown in Algorithm 2.

The bitand operation is used to extract the most significant bit plane T_1 of the original image F, and then the image is scanned pixel by pixel. The chaotic sequence C_h is used as the chaotic input sequence to participate in the encryption process of each pixel. When the most significant bits $T_1(i, j)$ and $T_1(i, j + 1)$ of the adjacent pixels are both 1, the latter and the chaotic sequence $C_h(i, j)$ are summed, 128 modulo operations are performed, and the most significant bit is changed to 0. When the

most significant bits $T_1(i, j)$ and $T_1(i, j + 1)$ of the adjacent pixels are both 0, the latter and the value of the chaotic sequence $C_h(i, j)$ are summed and 128 modulo operations are performed. At the same time, the most significant bit is changed to 1. Similarly, we use C_v and C_d to perform analogous operations in the vertical and diagonal directions. The final encrypted image E is obtained.

Algorithm 2： Neighbor bit deprivation

Input： F, sequences C_h
Output： encrypted image E
$m, n \leftarrow$ size (F), $T_1 \leftarrow$ bitand (F, 128) / 128
for $i \leftarrow 1$ to m **do**
  for $j \leftarrow 1$ to n **do**
    if $T_1(i, j) = T_1(i, j + 1) = 1$ **then**
      $E(i, j + 1) \leftarrow \text{mod}(T_1(i, j + 1) + C_h(i, j), 128) + 128$
    else if $T_1(\text{i, j}) = T_1(\text{i, j} + 1) = 0$ **then**
      $E(i, j + 1) \leftarrow \text{mod}(T_1(i, j + 1) + C_h(i, j), 128)$
    end
  end
end

4.3. Decryption algorithms

The chaotic sequences in the decryption process are identical to those in the encryption process, making the techniques reversible. Thus, the decryption procedure of the proposed algorithm reverses the encryption process. Due to space constraints, the decryption process is not described in detail.

5. Experimental results and algorithmic security analysis

5.1. Experimental results

5.1.1. Encryption and decryption results

In this paper, “Lena”, “Peppers”, and “Baboon” of size 512 × 512 were selected from the standard test images. As shown in Figure 10, the original image (Column a) contains emerging textures and edge details. The encrypted image (Column b) shows a uniform noise distribution without any recognizable structural features, indicating that the algorithm effectively destroys the visual information of the image. The decrypted image (Column c) recovers the details of the original image intact.

5.1.2. Histograms and their analysis

A histogram visually represents the distribution of pixels in an image, indicating the frequency of each gray level. For encrypted images, ensuring that various gray values correspond to approximately equal gray frequencies demonstrates strong resistance to statistical analysis attacks, as the visual representation is a flat histogram. As depicted in Figure 11, the histograms of the different encrypted images proposed in this paper are all notably flat.

(a) (b) (c)

Figure 10. Encryption and decryption of test images.

5.2. Statistical analysis

5.2.1. Pixel correlation analysis

The correlation between neighboring pixel points measures the degree of similarity between neighboring pixels, including horizontal, vertical, and diagonal neighboring pixel correlation. If the correlation between neighboring pixels is high, the encryption algorithm can be easily attacked by statistical data. Therefore, a good encryption algorithm should ensure that the neighboring pixels of an encrypted image have the lowest possible correlation between them. The formula for calculating the correlation of neighboring pixels is shown in Eq (16).

$$\begin{cases} E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \\ \mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \\ D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \\ r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \end{cases} \tag{16}$$

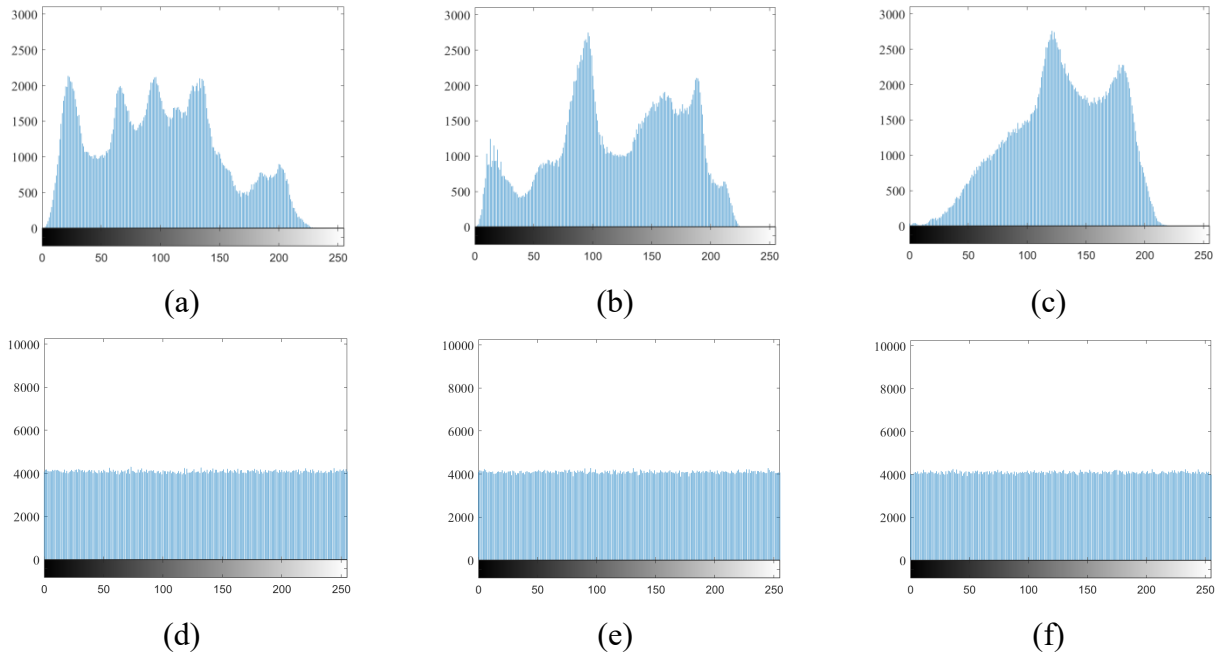


Figure 11. (a)–(c) Histograms of the original Lena, Peppers, and Baboon images. (d)–(f) Histograms of the encrypted Lena, Peppers, and Baboon images.

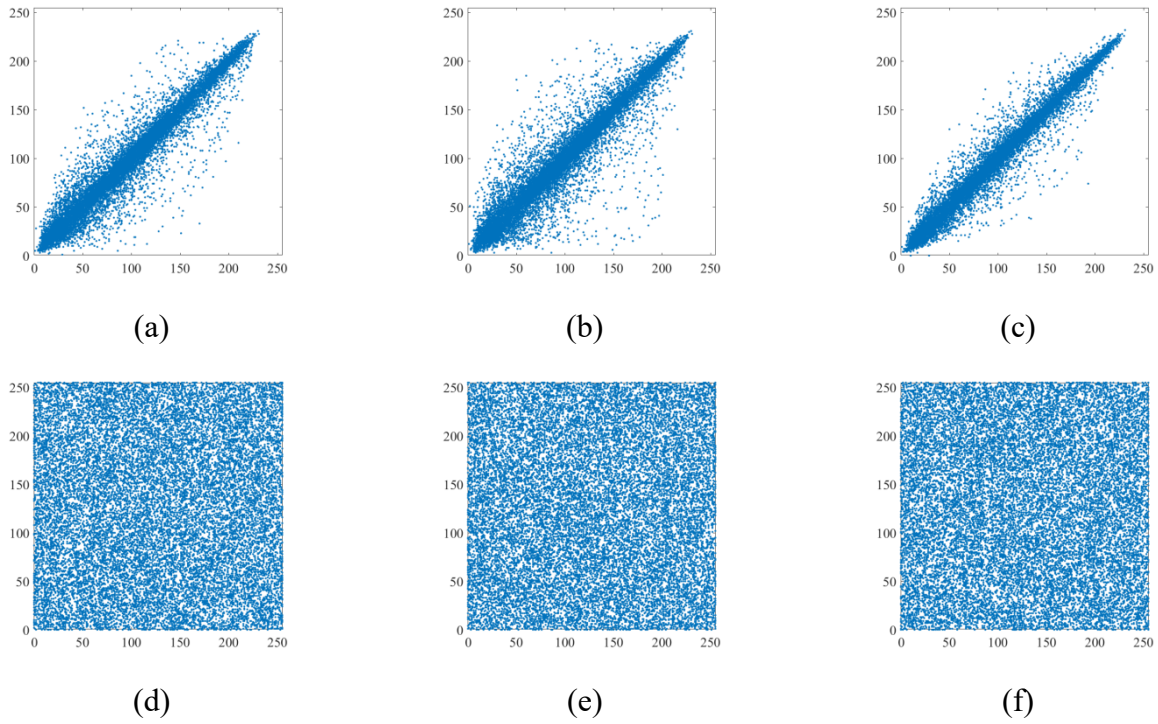


Figure 12. Pixel correlation of Lena. (a)–(c) Correlations of the original image in the horizontal, vertical, and diagonal directions. (d)–(f) Correlations of the encrypted image in the horizontal, vertical, and diagonal directions.

Here, x and y are the gray values of two neighboring pixels and N is the total number of pixels

selected from the image. We randomly selected 5000 pairs of horizontal neighboring pixels from the image and calculated the correlation coefficient between each pair. Figure 12 shows the histogram distribution between neighboring pixels of the Lena image.

Furthermore, for a more intuitive description, we computed the data of the neighboring pixel correlations in different images and their corresponding encrypted images separately, which is shown in Table 3.

Table 3. Correlation coefficients of test images.

Image	Orientation	CC	Proposed	[36]	[37]	[38]	[39]	[40]
Lena	H	0.9742	0.0005	–0.0049	0.0013	0.0073	0.0083	0.0021
	V	0.9857	–0.0012	0.0070	0.0085	0.0115	–0.0021	0.0010
	D	0.9607	0.0005	–0.0013	0.0004	–0.0012	–0.0025	–0.0023
Peppers	H	0.9773	0.0001	–0.0054	–0.0016	/	0.0067	–0.0008
	V	0.9788	0.0027	0.0012	0.0018	/	–0.0050	0.0014
	D	0.9644	0.0013	0.0072	–0.0022	/	–0.0059	0.0004
Baboon	H	0.8638	0.0041	0.0093	0.0011	–0.0086	/	–0.0014
	V	0.7584	–0.0004	–0.0052	–0.0060	0.0056	/	–0.0007
	D	0.7267	–0.0012	–0.0004	0.0005	0.0092	/	0.0026

5.2.2. Differential attack analysis

An attacker can defeat the encryption algorithm by changing the pixel values of the plain image and comparing the pixel values of the two encrypted images. To withstand a differential attack, the encrypted image should change significantly when we modify the pixel values of the plain image.

To assess the impact of changing individual pixel values in a simple image, we frequently employ two tests: NPCR (number of pixel change rates) and UACI (uniform average change intensity). The formulas are shown in Eqs (17)–(19).

$$D(i,j)=\begin{cases}0, & c_1(i,j)=c_2(i,j)\\ 1, & c_1(i,j)\neq c_2(i,j)\end{cases} \tag{17}$$

$$NPCR=\frac{\sum_{i,j}D(i,j)}{M\times N}\times 100 \tag{18}$$

$$UACI=\frac{1}{M\times N}\left[\sum_{i,j}\frac{|c_1(i,j)-c_2(i,j)|}{255}\right]\times 100 \tag{19}$$

where $c_1(i,j)$ and $c_2(i,j)$ are two pixels at the same position (i,j) of the two images. Theoretically, the expected values of NPCR and UACI are 99.6093% and 33.4635%, respectively. Statistical tests can verify whether the NPCR and UACI results follow the expected distribution, and a significance level of $\alpha = 0.05$ was set in this experiment.

The critical value of NPCR for the significance level α is labeled as N_α^*, which can be calculated by Eq (20), where $\phi^{-1}(\alpha)$ is the inverse cumulative density function of the standard normal distribution

N (0, 1). The UACI lower limit of significance level α is labeled as U_α^{*-} and the upper limit is labeled as U_α^{*+}, and can be calculated by Eqs (21)–(23).

$$N_\alpha^* = \frac{L - \phi^{-1}(\alpha)\sqrt{L/MN}}{L+1} \tag{20}$$

$$\begin{cases} U_\alpha^{*-} = \mu_u - \phi^{-1}(\alpha/2) \times \sigma_u \\ U_\alpha^{*+} = \mu_u + \phi^{-1}(\alpha/2) \times \sigma_u \end{cases} \tag{21}$$

$$\mu_u = \frac{L+2}{3L+3} \tag{22}$$

$$\sigma_u^2 = \frac{(L+2)(L^2+2L+3)}{18MNL(L+1)^2} \tag{23}$$

Table 4. Average NPCR and comparison with other algorithms.

Size	Image	Proposed	[39]	[41]	[42]	[43]	[44]	[45]
256 × 256	5.1.09	**99.6071**	99.6109	99.6093	99.6042	99.5900	99.5796	99.611
	5.1.10	**99.6044**	99.5834	99.6095	99.5447	99.5393	99.6025	99.652
	5.1.11	**99.6029**	99.6155	99.6133	99.7993	66.6535	99.6183	99.611
	5.1.12	**99.5972**	99.6170	99.6123	99.6879	99.5749	99.5925	99.574
	5.1.13	**99.6014**	99.5850	99.6050	99.5838	99.5900	99.6138	99.630
	5.1.14	**99.5958**	99.6033	99.6110	99.6861	99.5789	99.6101	99.587
512 × 512	5.2.08	**99.6040**	99.6037	99.6070	99.6271	99.6497	99.5922	99.596
	5.2.09	**99.5984**	99.6063	99.6106	99.5834	99.6116	99.6145	33.607
	5.2.10	**99.6006**	99.6174	99.6096	99.6254	99.5801	99.6034	99.616
	7.1.01	**99.6051**	99.6212	99.6095	99.6152	99.6288	99.5839	99.611
	7.1.02	**99.6124**	99.6059	99.6117	99.6218	99.6535	99.6651	99.620
	7.1.03	**99.6066**	99.6071	99.6123	99.5835	99.5909	99.5776	99.601
	7.1.04	**99.6005**	99.5991	99.6114	99.5661	99.5930	99.6184	99.616
	7.1.05	**99.5911**	99.6104	99.6099	99.6942	99.5609	99.6255	99.614
	7.1.06	**99.6120**	99.6193	99.6064	99.5670	99.5922	99.6262	99.588
	7.1.07	**99.5915**	99.6109	99.6068	99.5953	99.5911	99.6155	99.621
	7.1.08	**99.6070**	99.6044	99.6097	99.5845	99.5703	99.5884	99.609
	7.1.09	**99.5989**	99.5953	99.6112	99.6317	99.6029	99.6002	99.616
	7.1.10	99.5076	99.6067	99.6096	99.6182	99.5093	99.6388	99.602
	boat.512	**99.6025**	99.6059	99.6084	99.5718	99.5952	99.6154	99.595
	gray.512	**99.5995**	99.6315	99.6074	99.6275	99.5907	99.6027	99.608
	ruler.512	**99.6001**	99.6017	99.6092	99.6365	99.5895	99.6183	99.608
1024 × 1024	5.3.01	**99.6017**	99.6075	99.6095	99.5974	99.5596	99.6621	/
	5.3.02	**99.6036**	99.6074	99.6095	99.6649	99.6007	99.5685	/
	7.2.01	**99.6086**	99.6099	99.6092	99.5982	99.6639	99.5566	/

where L is the grayscale range. At a significance level of 0.05, when the image size is 256 × 256, the threshold of N_α^* is 99.5693%, and the range of (U_α^{*-}, U_α^{*+}) is calculated as (33.2824%, 33.6447%). Similarly, when the image size is 512 × 512, the threshold of N_α^* is 99.5893%, and the range of (U_α^{*-}, U_α^{*+}) is (33.3730%, 33.5541%). When the image size is 1024 × 1024, the threshold of N_α^* is 99.5994% and the range of (U_α^{*-}, U_α^{*+}) is (33.4183%, 33.5088%).

Tables 4 and 5 list the NPCR and UACI corresponding to different images and different algorithms. Theoretically, the encryption system is secure when the value of NPCR is greater than the threshold value and the value of UACI lies within the range. We mark the results in the tables that conform to the statistical test with bold type. From the observation, we can see that our proposed encryption system can pass the NPCR and UACI tests, indicating that it is very resistant to differential attacks.

Table 5. Average UACI and comparison with other algorithms.

Size	Image	Proposed	[39]	[41]	[42]	[43]	[44]	[45]
256 × 256	5.1.09	**33.3989**	33.5967	33.4723	33.552	33.5214	33.4456	33.545
	5.1.10	**33.4034**	33.5314	33.4663	33.453	33.4215	33.4946	33.448
	5.1.11	**33.4324**	33.5570	33.4554	33.586	33.4014	33.5541	33.490
	5.1.12	**33.4150**	33.4695	33.4604	33.453	33.4158	33.4302	33.518
	5.1.13	**33.4733**	33.4783	33.4601	33.520	33.4236	33.4438	33.363
	5.1.14	**33.4245**	33.5192	33.4606	33.440	33.3951	33.4655	33.379
512 × 512	5.2.08	**33.4264**	33.4948	33.4734	33.692	33.3958	33.4008	33.457
	5.2.09	**33.4301**	33.4416	33.4572	33.548	33.4182	33.4804	33.439
	5.2.10	**33.4387**	33.5077	33.4575	33.454	33.4263	33.4563	33.512
	7.1.01	**33.4525**	33.4472	33.4726	33.648	33.4474	33.5037	33.417
	7.1.02	**33.4477**	33.4678	33.4563	33.465	33.4326	33.4237	33.412
	7.1.03	**33.4459**	33.3742	33.4535	33.273	33.4836	33.4291	33.471
	7.1.04	**33.4061**	33.4534	33.4475	33.202	33.4782	33.4739	33.390
	7.1.05	**33.4185**	33.5153	33.4559	33.830	33.4716	33.4362	33.525
	7.1.06	**33.4098**	33.4478	33.4515	33.627	33.4365	33.3954	33.470
	7.1.07	**33.4335**	33.4815	33.4638	33.609	33.4313	33.4073	33.383
	7.1.08	**33.4659**	33.4920	33.4536	33.375	33.4460	33.4332	33.448
	7.1.09	**33.4355**	33.5357	33.4729	33.530	33.3856	33.4177	33.408
	7.1.10	**33.4027**	33.4210	33.4605	33.438	33.3941	33.4344	33.478
	boat.512	**33.4555**	33.4441	33.4434	33.374	33.3973	33.4654	33.475
	gray.512	**33.4500**	33.4402	33.4588	33.507	33.4089	33.4608	33.526
	ruler.512	**33.4373**	33.4734	33.4637	33.415	33.4635	33.4262	33.407
1024 × 1024	5.3.01	**33.4394**	33.4791	33.4511	33.508	33.4392	33.4585	/
	5.3.02	**33.4395**	33.4752	33.4536	33.514	33.4547	33.4605	/
	7.2.01	**33.4494**	33.4444	33.4606	33.487	33.4301	33.4556	/

5.3. Information entropy

Information entropy is a statistical indicator that reflects the randomness of the information. To assess the randomness, the information entropy of the encrypted image is calculated by Eq (24).

$$H(s)=\sum_{i=0}^{2^{L}+1} p(s_i)\log_2 \frac{1}{p(s_i)} \tag{24}$$

where $p(s_i)$ denotes the probability of s_i occurring.

In theory, the closer the value of information entropy is to 8, the higher the pixel clutter and the less likely information leakage is. Table 6 compares the information entropy of various test images with that in other papers.

Table 6. Information entropy of images.

Size	Image	Entropy	Proposed	[46]	[41]	[47]
256 × 256	5.1.09	6.709312	**7.999324**	7.9027	7.9971	7.9966
	5.1.10	7.311807	**7.999331**	7.9022	7.9974	7.9971
	5.1.11	6.452275	**7.999375**	7.9016	7.9969	7.9975
	5.1.12	6.705667	**7.999395**	7.9023	7.9972	7.9972
	5.1.13	1.548314	**7.999324**	7.9036	7.9969	7.9965
	5.1.14	7.342433	**7.999389**	7.9002	7.9974	7.9977
512 × 512	5.2.08	7.201008	**7.999816**	7.9025	7.9993	7.9991
	5.2.09	6.993994	**7.999807**	7.9027	7.9993	7.9992
	5.2.10	5.705560	**7.999819**	7.9021	7.9993	7.9991
	7.1.01	6.027415	**7.999820**	7.9027	7.9991	7.9990
	7.1.02	4.004499	**7.999814**	7.8935	7.9992	7.9991
	7.1.03	5.465740	**7.999810**	7.9007	7.9993	7.9990
	7.1.04	6.107418	**7.999827**	7.9022	7.9993	7.9992
	7.1.05	6.563196	**7.999809**	7.9022	7.9992	7.9992
	7.1.06	6.695283	**7.999795**	7.9031	7.9993	7.9992
	7.1.07	5.991599	**7.999805**	7.9028	7.9993	7.9991
	7.1.08	5.053448	**7.999808**	7.9024	7.9993	7.9990
	7.1.09	6.189814	**7.999823**	7.9028	7.9992	7.9991
	7.1.10	5.908790	**7.999808**	7.9027	7.9993	7.9990
	boat.512	7.191370	**7.999837**	7.9025	7.9994	7.9992
	gray.512	4.392295	**7.999825**	7.8871	7.9994	7.9993
	ruler.512	0.500033	**7.999815**	7.8987	7.9992	7.9987
1024 × 1024	5.3.01	7.523737	**7.999953**	7.9025	7.9998	7.9998
	5.3.02	6.830330	**7.999950**	7.9025	7.9998	7.9998
	7.2.01	5.641454	**7.999955**	7.9019	7.9998	7.9998

5.4. Robustness analysis

Robustness is a key criterion for assessing the immunity of an encryption system to interference. The image may lose some information during the encryption process or be attacked by noise. As a result, the algorithm must be highly resilient.

The MSE (mean squared error) evaluates the viability of the proposed scheme by comparing the pixels of the original and decrypted images. The PSNR (peak signal to noise ratio) is also employed

as a metric to assess the quality of the image encryption system. They are indicated in Eq (25).

$$\begin{cases} \text{MSE} = \dfrac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(P(i,j)-D(i,j)\right)^2}{M\times N} \\ \text{PSNR} = 10\log\left(\dfrac{I_{\max}^2}{\text{MSE}}\right) \end{cases} \tag{25}$$

where $P(i, j)$ and $D(i, j)$ represent the pixels of the original and decrypted image, and I_{max} is 255. In theory, a robust encryption system should result in a higher PSNR number. Table 7 compares the PSNR values of various cropping attacks with those found in other papers. It can be shown that our proposed scheme is more resilient and secure than others.

Table 7. PSNR between the decrypted image and original image under cropping attacks.

Image	Crop	Proposed	[48]	[49]	[50]
Lena	1/16	**23.5171**	20.4047	21.2741	20.3857
	1/4	**16.1752**	14.4521	15.2478	14.3735
	1/2	**14.9706**	11.4674	12.2889	11.3865
Baboon	1/16	**22.4511**	21.3753	22.2541	21.2852
	1/4	15.2670	15.3844	16.2457	15.3401
	1/2	**13.1846**	12.3608	13.2547	12.3520
Peppers	1/16	**20.6585**	20.2985	20.2241	20.3598
	1/4	**16.1117**	14.3057	14.1524	14.4016
	1/2	**14.6665**	11.3580	11.2952	11.3797

Table 8. PSNR between the decrypted image and original image under noise attacks.

Image	Noise	Proposed	[48]	[49]	[50]
Lena	0.005	30.9880	31.2751	31.5649	31.5876
	0.05	**21.2840**	21.2502	21.3198	21.3575
	0.01	**18.5469**	18.3147	18.2658	18.3675
Baboon	0.005	**31.5319**	32.4933	31.3731	32.2013
	0.05	21.1004	22.2653	21.2675	22.2380
	0.01	**19.3382**	19.2684	18.3223	19.2031
Peppers	0.005	30.2924	31.2206	30.1426	31.0920
	0.05	**20.4851**	21.2503	20.3632	21.2046
	0.01	17.9136	18.2231	17.4689	18.2498

Table 8 compares the PSNR values of the encrypted image Lena after adding salt and pepper noise of various intensities with the results reported in other papers. The findings show that even after adding noise with unequal density to the encrypted image Lena, the associated values generated by this technique meet the requirements, demonstrating the algorithm's robustness.

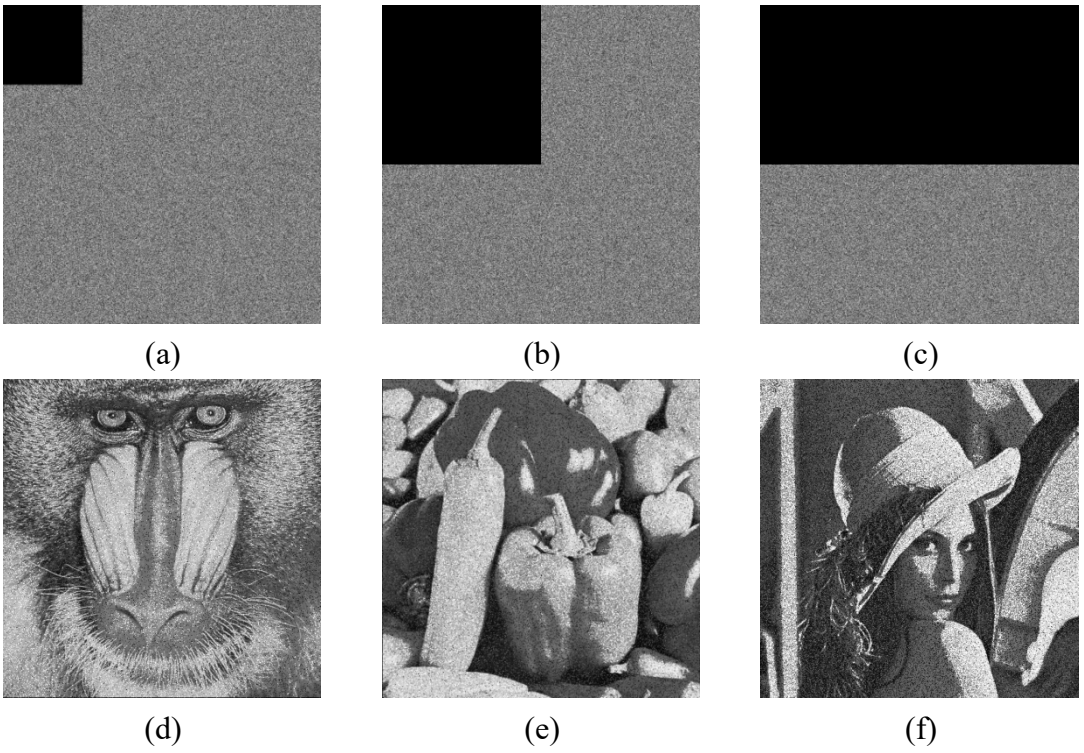


Figure 13. Cropping attack results. (a)–(c) Cropping attack of the encrypted images. (d)–(f) The decrypted images of (a)–(c).

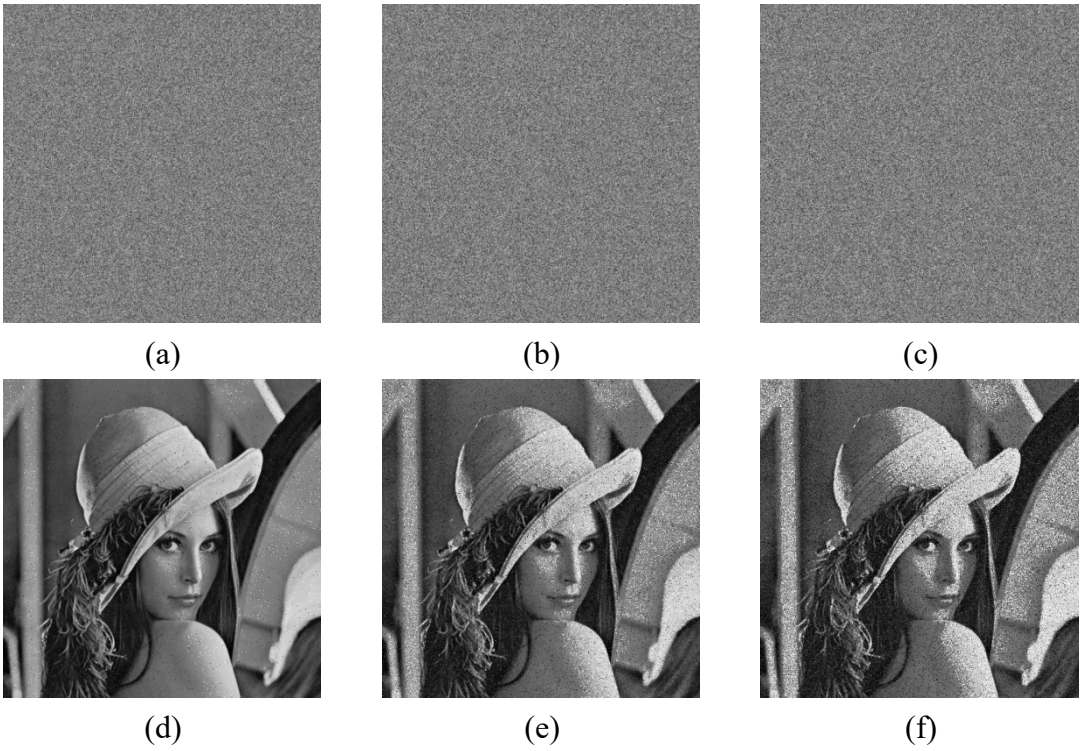


Figure 14. Noise attack results. (a)–(c) Noise attack of the encrypted images with a density of 0.005, 0.05, and 0.1. (d)–(f) the decrypted image of (a)–(c).

As shown in Figure 13, this study tested the encryption system with rigorous cropping attacks, and simulated the recovery of the ciphertext image after it was cropped by 1/16, 1/4, and 1/2 of the area, respectively. The experimental results show that the proposed encryption system maintains excellent image recovery performance even under severe data loss. Pepper and salt noise with an intensity of 0.005, 0.05, and 0.1 is added to the encrypted image. The encrypted image with added noise is decrypted using the key. The decrypted image is shown in Figure 14. It can be directly observed that the encrypted image after adding pretzel noise can still be decrypted using the original key, and the decrypted image can be almost recovered as the original image.

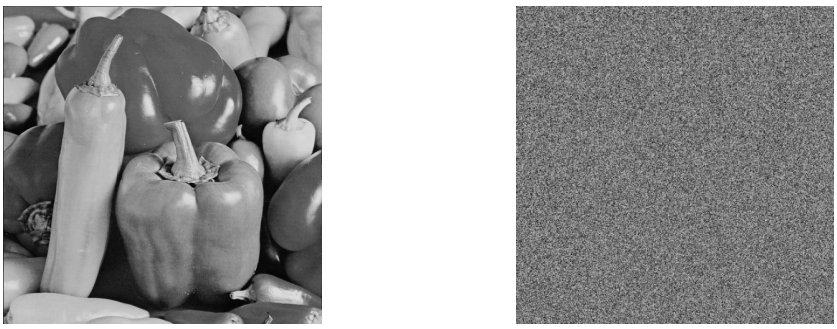
The results show that the encryption scheme proposed in this study can basically recover the information of the original image even if half of the encrypted image information is lost, and it also has good resistance under salt and pepper noise attack. Therefore, the algorithm proposed in this research is relatively robust.

5.5. Encryption time

In a series of standards to measure the quality of encryption algorithms, security analysis is essential, but the calculation time of the algorithm is also a factor worthy of attention. The shorter the encryption time, the higher the time efficiency of the algorithm. Table 9 lists the encryption time of this experiment and a comparison with other literature. It can be seen that the proposed algorithm not only meets the security analysis standard, but also has a short encryption time and is easy to implement.

Table 9. Encryption time and comparison with other algorithms.

	Proposed	[51]	[52]	[53]	[54]
Encryption time	2.553803	2.750966	2.522390	2.582389	2.443281



(a) (b)

Figure 15. Decrypted image of the key. (a) Decrypted image without changing. (b) Decrypted image with one bit changing.

5.6. Key space and sensitivity analysis

The key space is one of the most important characteristics for evaluating the strength of an encryption algorithm. The size of the key space has a direct impact on the algorithm's ability to

withstand brute-force attacks; the larger the key space, the more difficult it is to break. The key space we proposed is 2^{512}, which far exceeds the security threshold of 2^{128}, and therefore has a sufficient security guarantee.

Key sensitivity is a further indicator of the encryption scheme's robustness, as a minor change in the key might create an entirely different encryption. As shown in Figure 15, when only one bit of the decryption key is changed, the decryption result becomes a completely unrecognizable image. This experimental demonstrates that the proposed encryption scheme has excellent key sensitivity properties.

5.7. Chosen plaintext attack analysis

The chosen plaintext attack is one of four common types of attack. If an encryption system can withstand the chosen plaintext attack, it is likely that it can also resist the other three attacks. In our proposed technique, the encryption process is tightly associated with the original image and is extremely sensitive to minor changes in the plaintext, providing a strong defense against chosen plaintext attacks. Figure 16 shows two special plain images in all-black and all-white (Column a) and the associated encrypted images (Column b) with histograms (Column c). The entropy of the two images is 7.999820 and 7.999839. This means that the attacker cannot extract any meaningful information from the encrypted image, making the proposed approach very resistant to chosen plaintext attacks.

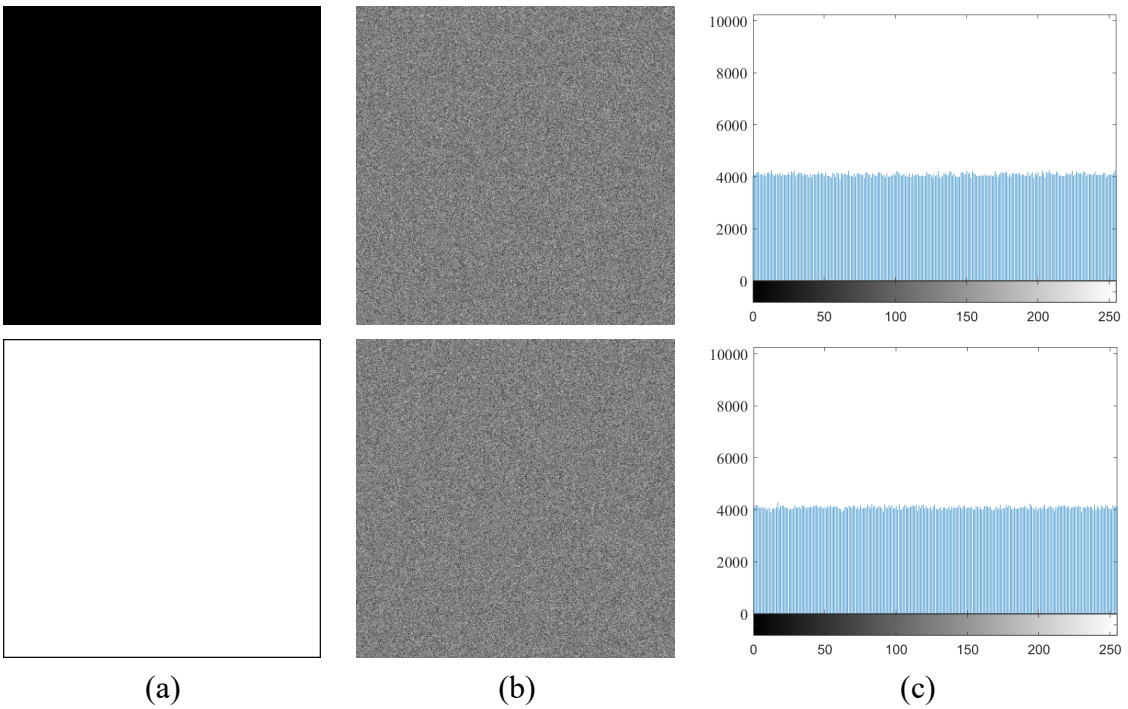


Figure 16. Test of all-black and all-white images against chosen plaintext attacks.

6. Conclusions

The current image encryption algorithms suffer from the challenges of the low complexity

inherent in spatial domain encryption and weak resistance to statistical attacks. To address such issues, this study proposes a unique image encryption system based on joint spatial-frequency domain processing. The strategy utilizes optical transformation to analyze the frequency domain data and combines innovative hash scrambling and bit diffusion algorithms to create a multilevel encryption system. Specifically, a chaotic system combined with hash scrambling is used in the space domain to replace the traditional scrambling method, and then the image is converted to the frequency domain by the Fourier transform, which is combined with the DNA coding technique and a bit deprivation algorithm to further enhance the diffusion effect. To evaluate the effectiveness of this study, a system security analysis is conducted to confirm that both the NPCR and UACI of this study outperform those of existing schemes. The experimental results demonstrate that the proposed innovative bit-level transform mechanism significantly enhances the diffusion diversity, and the collaborative encryption scheme in the spatial frequency domain offers robust security performance. However, the computational delay in the hardware implementation of the spatial-frequency domain transform needs to be further optimized. Future research can be explored in the field of multimodal message encryption. This study provides a new technical path for future research in image encryption.

Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the Natural Science Foundation of Fujian Province (Grant Nos 2024J01821 and 2024J01820), the Model Courses of Ideological and Political Education in Graduate Programs at Minnan Normal University (Grant No. SZ202406), the Research Project on Postgraduate Education and Teaching Reform of Minnan Normal University (Grant No. YJG202316), and the Principal Foundation of Minnan Normal University (Grant No. KJ18010).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. K. U. Shahna, A. Mohamed, A novel image encryption scheme using both pixel level and bit level permutation with chaotic map, *Appl. Soft Comput.*, **90** (2020), 106162. https://doi.org/10.1016/j.asoc.2020.106162
2. W. Feng, K. Zhang, J. Zhang, X. Zhao, Y. Chen, B. Cai, et al., Integrating fractional-order hopfield neural network with differentiated encryption: Achieving high-performance privacy protection for medical images, *Fractal Fract.*, **9** (2025), 426. https://doi.org/10.3390/fractalfract9070426
3. C. Ye, S. Tan, J. Wang, L. Shi, Q. Zuo, W. Feng, Social image security with encryption and watermarking in hybrid domains, *Entropy*, **27** (2025), 276. https://doi.org/10.3390/e27030276

4. W. Feng, J. Zhang, Y. Chen, Z. Qin, Y. Zhang, M. Ahmad, et al., Exploiting robust quadratic polynomial hyperchaotic map and pixel fusion strategy for efficient image encryption, *Expert Syst. Appl.*, **246** (2024), 123190. https://doi.org/10.1016/j.eswa.2024.123190
5. H. Li, S. Yu, W. Feng, Y. Chen, J. Zhang, Z. Qin, et al., Exploiting dynamic vector-level operations and a 2D-enhanced logistic modular map for efficient chaotic image encryption, *Entropy*, **25** (2023), 1147. https://doi.org/10.3390/e25081147
6. Z. Hua, J. Yao, Y. Zhang, H. Bao, S. Yi, Two-dimensional coupled complex chaotic map, *IEEE Trans. Ind. Inf.*, **21** (2025), 85–95. https://doi.org/10.1109/TII.2024.3431085
7. B. Abd-El-Atty, Efficient S-box construction based on quantum-inspired quantum walks with PSO algorithm and its application to image cryptosystem, *Complex Intell. Syst.*, **9** (2023), 4817–4835. https://doi.org/10.1007/s40747-023-00988-7
8. C. Wang, Y. Zhang, A novel image encryption algorithm with deep neural network, *Signal Process.*, **196** (2022), 108536. https://doi.org/10.1016/j.sigpro.2022.108536
9. H. Liu, Y. Xu, C. Ma, Chaos-based image hybrid encryption algorithm using key stretching and hash feedback, *Optik*, **216** (2020), 164925. https://doi.org/10.1016/j.ijleo.2020.164925
10. L. Li, A novel chaotic map application in image encryption algorithm, *Expert Syst. Appl.,* **35** (2024), 124316. https://doi.org/10.1016/j.eswa.2024.124316
11. W. Dai, B. Li, Q. Du, Z. Zhu, A. Liu, Chaos-based index-of-min hashing scheme for cancellable biometrics security, *IEEE Trans. Inf. Forensics Secur.*, **19** (2024), 8982–8997. https://doi.org/10.1109/TIFS.2024.3455109
12. C. Xue, H. Wan, P. Gu, N. Jiang, Y. Hong, Z. Zhang, Ultrafast secure key distribution based on random dna coding and electro-optic chaos synchronization, *IEEE J. Quantum Electron.*, **58** (2022). https://doi.org/10.1109/JQE.2021.3139711
13. A. K. Singh, K. Chatterjee, A. Singh, An image security model based on chaos and DNA cryptography for IIoT images, *IEEE Trans. Ind. Inf.*, **19** (2023), 1957–1964. https://doi.org/10.1109/TII.2022.3176054
14. X. Wang, H. Zhao, Y. Hou, C. Luo, Y. Zhang, C. Wang, Chaotic image encryption algorithm based on pseudo-random bit sequence and DNA plane, *Mod. Phys. Lett. B*, **33** (2019), 1950263. https://doi.org/10.1142/S0217984919502634
15. H. Nematzadeh, R. Enayatifar, M. Yadollahi, M. Lee, G. Jeong, Binary search tree image encryption with DNA, *Optik*, **202** (2020), 163505. https://doi.org/10.1016/j.ijleo.2019.163505
16. S. Zhang, L. Liu, A novel image encryption algorithm based on SPWLCM and DNA coding, *Math. Comput. Simul.*, **190** (2021), 723–744. https://doi.org/10.1016/j.matcom.2021.06.012
17. Z. Yang, Y. Cao, Y. Ji, Z. Liu, H. Chen, Securing color image by using bit-level modified integer nonlinear coupled chaos model in Fresnel diffraction domains, *Opt. Lasers Eng.*, **152** (2022), 106969. https://doi.org/10.1016/j.optlaseng.2022.106969
18. K. Qian, W. Feng, Z. Qin, J. Zhang, X. Luo, Z. Zhu, A novel image encryption scheme based on memristive chaotic system and combining bidirectional bit-level cyclic shift and dynamic DNA-level diffusion, *Front. Phys.*, **10** (2022), 963795. https://doi.org/10.3389/fphy.2022.963795
19. J. Gao, Y. Wang, Z. Song, S. Wang, Quantum image encryption based on quantum DNA codec and pixel-level scrambling, *Entropy*, **25** (2023), 865. https://doi.org/10.3390/e25060865
20. A. A. A. El-Latif, B. Abd-El-Atty, Adaptive particle swarm optimization with quantum-inspired quantum walks for robust image security, *IEEE Access*, **11** (2023), 71143–71153. https://doi.org/10.1109/ACCESS.2023.3286347

21. Y. Su, X. Wang, H. Gao, Chaotic image encryption algorithm based on bit-level feedback adjustment, *Inf. Sci.*, **679** (2024), 121088. https://doi.org/10.1016/j.ins.2024.121088
22. X. Jiang, Y. Xiao, Y. Xie, B. Liu, Y. Ye, T. Song, et al., Exploiting optical chaos for double images encryption with compressive sensing and double random phase encoding, *Opt. Commun.*, **484** (2021), 126683. https://doi.org/10.1016/j.optcom.2020.126683
23. T. Zeng, Y. Zhu, E.Y. Lam, Deep learning for digital holography: a review, *Opt. Express*, **29** (2021), 40572. https://doi.org/10.1364/OE.443367
24. Y. Hualong, G. Daidou, Research on double encryption of ghost imaging by SegNet deep neural network, *IEEE Photonics Technol. Lett.*, **36** (2024), 669–672. https://doi.org/10.1109/LPT.2024.3379554
25. Y. Wang, Y. Su, X. Sun, X. Hao, Y. Liu, X. Zhao, et al., Principle and implementation of stokes vector polarization imaging technology, *Appl. Sci.*, **12** (2022), 6613. https://doi.org/10.3390/app12136613
26. L. Wang, T. Wang, R. Yan, X. Yue, H. Wang, Y. Wang, et al., Color printing and encryption with polarization-switchable structural colors on all-dielectric metasurfaces, *Nano Lett.*, **23** (2023), 5581–5587. https://doi.org/10.1021/acs.nanolett.3c01007
27. J. Zhang, Z. Huang, X. Li, M. Wu, X. Wang, Y. Dong, Quantum image encryption based on quantum image decomposition, *Int. J. Theor. Phys.*, **60** (2021), 2930–2942. https://doi.org/10.1007/s10773-021-04862-5
28. K. Benyahia, A. Khobzaoui, S. Benbakreti, Hybrid image encryption: leveraging DNA sequencing and Lorenz chaotic dynamics for enhanced security, *Cluster Comput*., **28** (2025), 218. https://doi.org/10.1007/s10586-024-04948-9
29. C. Zhang, J. Cheng, D. Chen, Cryptanalysis of an image encryption algorithm based on a 2D hyperchaotic map, *Entropy*, **24** (2022), 1551. https://doi.org/10.3390/e24111551
30. J. Tang, Z. Zhang, T. Huang, Two-dimensional cosine–sine interleaved chaotic system for secure communication, *IEEE Trans. Circuits Syst. II Express Briefs*, **71** (2024), 2479–2483. https://doi.org/10.1109/TCSII.2023.3337145
31. A. Yahi, B. Tewfik, M. E. H. Daachi, N. Diffellah, A color image encryption scheme based on 1D cubic map, *Optik*, **249** (2022), 168290. https://doi.org/10.1016/j.ijleo.2021.168290
32. S. M. Basha, P. Mathivanan, A. B. Ganesh, Bit level color image encryption using Logistic-Sine-Tent-Chebyshev (LSTC) map, *Optik*, **259** (2022), 168956. https://doi.org/10.1016/j.ijleo.2022.168956
33. M. T. Yassen, Chaos control of Chen chaotic dynamical system, *Chaos, Solitons Fractals*, **15** (2003), 271–283. https://doi.org/10.1016/s0960-0779(01)00251-x
34. F. Pengfei, L. Miaomiao, L. Min, L. Han, Image encryption algorithm based on hyperchaotic system and DNA coding, in *2021 International Conference on Computer Communication and Artificial Intelligence (CCAI)*, *IEEE*, (2021), 41–46. https://doi.org/10.1109/CCAI50917.2021.9447470
35. H. Wen, S. Kang, Z. Wu, Y. Lin, Y. Huang, Dynamic RNA coding color image cipher based on chain feedback structure, *Mathematics*, **11** (2023), 3133. https://doi.org/10.3390/math11143133
36. Y. Huang, Q. Zhang, Y. Zhao, Color image encryption algorithm based on hybrid chaos and layered strategies, *J. Inf. Secur. Appl.*, **89** (2025), 103921. https://doi.org/10.1016/j.jisa.2024.103921

37. M. Wang, X. Song, S. Liu, X. Zhao, N. Zhou, A novel 2D Log-Logistic–Sine chaotic map for image encryption, *Nonlinear Dyn.*, **113** (2025), 2867–2896. https://doi.org/10.1007/s11071-024-10331-5
38. R. Zhang, R. Zhou, J. Luo, Nonequal-length image encryption based on bitplane chaotic mapping, *Sci. Rep.*, **14** (2024), 9075. https://doi.org/10.1038/s41598-024-58612-8
39. X. Wang, S. Chen, Y. Zhang, A chaotic image encryption algorithm based on random dynamic mixing, *Opt. Laser Techol.*, **138** (2021), 106837. https://doi.org/10.1016/j.optlastec.2020.106837
40. X. An, S. Liu, L. Xiong, J. Zhang, X. Li, Mixed gray-color images encryption algorithm based on a memristor chaotic system and 2D compression sensing, *Expert Syst. Appl.*, **243** (2024), 122899. https://doi.org/10.1016/j.eswa.2023.122899
41. Y. Xian, X. Wang, Fractal sorting matrix and its application on chaotic image encryption, *Inf. Sci.*, **547** (2021), 1154–1169. https://doi.org/10.1016/j.ins.2020.09.055
42. Q. Cun, X. Tong, Z. Wang, M. Zhang, A new chaotic image encryption algorithm based on dynamic DNA coding and RNA computing, *Visual Comput.*, **39** (2023), 6589–6608. https://doi.org/10.1007/s00371-022-02750-5
43. K. M. Hosny, S. T. Kamal, M. M. Darwish, A novel color image encryption based on fractional shifted Gegenbauer moments and 2D logistic-sine map, *Visual Comput.*, **39** (2023), 1027–1044. https://doi.org/10.1007/s00371-021-02382-1
44. Q. Lai, Y. Liu, L. Yang, Remote sensing image encryption algorithm utilizing 2D Logistic memristive hyperchaotic map and SHA-512, *Sci. China Technol. Sci.*, **67** (2024), 1553–1566. https://doi.org/10.1007/s11431-023-2584-y
45. X. Liu, X. Tong, M. Zhang, Z. Wang, A highly secure image encryption algorithm based on conservative hyperchaotic system and dynamic biogenetic gene algorithms, *Chaos, Solitons Fractals*, **171** (2023), 113450. https://doi.org/10.1016/j.chaos.2023.113450
46. A. Moatsum, S. Azman, S. T. Je, S. A. Rami, A new hybrid digital chaotic system with applications in image encryption, *Signal Process.*, **160** (2019), 45–58. https://doi.org/10.1016/j.sigpro.2019.02.016
47. Y. Zhou, L. Bao, C. L. P. Chen, Image encryption using a new parametric switching chaotic system, *Signal Process.*, **93** (2013), 3039–3052. https://doi.org/10.1016/j.sigpro.2013.04.021
48. T. Hu, Y. Liu, L. H. Gong, C. J. Ouyang, An image encryption scheme combining chaos with cycle operation for DNA sequences, *Nonlinear Dyn.*, **87** (2017), 51–66. https://doi.org/10.1007/s11071-016-3024-6
49. S. Kumar, D. Sharma, A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm, *Artif. Intell. Rev.*, **57**(2024), 87. https://doi.org/10.1007/s10462-024-10719-0
50. S. Kumar, D. Sharma, Image scrambling encryption using chaotic map and genetic algorithm: a hybrid approach for enhanced security, *Nonlinear Dyn.*, **112** (2024), 12537–12564. https://doi.org/10.1007/s11071-024-09670-0
51. W. Alexan, M. Elkandoz, M. Mashaly, E. Azab, A. Aboshousha, Color image encryption through chaos and KAA map, *IEEE Access*, **11** (2023), 11541–11554. https://doi.org/10.1109/access.2023.3242311
52. H. Wen, Y. Lin, S. Kang, X. Zhang, K. Zou, Secure image encryption algorithm using chaos-based block permutation and weighted bit planes chain diffusion, *IScience*, **27** (2024), 108610. https://doi.org/10.1016/j.isci.2023.108610

53. W. Alexan, M. ElBeltagy, A. Aboshousha, Rgb image encryption through cellular automata, s-box and the lorenz system, *Symmetry*, **14** (2022), 443. https://doi.org/10.3390/sym14030443
54. M. A. B. Farah, A. Farah, T. Farah, An image encryption scheme based on a new hybrid chaotic map and optimized substitution box, *Nonlinear Dyn.*, **99** (2020), 3041–3064. https://doi.org/10.1007/s11071-019-05413-8

AIMS Press