PAPER

Frequency-domain image encryption based on IWT and 3D S-box

Physica Scripta

PAPER

Frequency-domain image encryption based on IWT and 3D S-box

Heping Wen[1,*], Zhaoyang Feng[1], Chixin Bai[1], Yiting Lin[1], Xiangyu Zhang[2] and Wei Feng[3]

[1] University of Electronic Science and Technology of China, Zhongshan Institute, Zhongshan 528402, People's Republic of China
[2] School of Automation, Guangdong University of Technology, Guangzhou 510006, People's Republic of China
[3] School of Mathematics and Computer Science, Panzhihua University, Panzhihua 617000, People's Republic of China
[*] Author to whom any correspondence should be addressed.

E-mail: wenheping@uestc.edu.com, 202102021045@stu.zsc.edu.cn, 202101004038@stu.zsc.edu.cn, dr.yitinglin@gmail.com, 2112304329@mail2.gdut.edu.cn and fengwei@pzhu.edu.cn

RECEIVED
2 November 2023

REVISED
28 February 2024

ACCEPTED FOR PUBLICATION
6 March 2024

PUBLISHED
18 April 2024

Keywords: image encryption, chaotic encryption, privacy-preserving, 3D S-box, integer wavelet transform

Abstract

Most of the existing spatial domain image encryption techniques suffer from the difficulty of resisting cryptographic attacks. For this reason, this paper proposes a frequency-domain based digital image encryption scheme by combining Integer Wavelet Transform (IWT), three-dimensional S-box and chaotic system. First, the plaintext image is decomposed into different frequency subbands by IWT to map the digital image from spatial domain to frequency domain. Second, the plaintext hash value is selected as the dynamic key, and dynamic chaotic pseudo-random sequences with associations are generated, which are used for the encryption of each module respectively. Then, a three-dimensional S-box is designed to encrypt the information-rich low-frequency information using 'bit-permutation three-dimensional S-box replace ciphertext interleaved diffusion', while the high-frequency information is encrypted using a lightweight 'XOR-row column permutation' operation. Finally, the secure ciphertext for public channel transmission is obtained by the reconstruction method. The scheme of this paper, the frequency domain transformation is implemented through IWT, which enhances the ability to resist attacks. In addition, the diffusion encryption modules employ the introduction of ciphertext interleaved diffusion and parallel encryption mechanisms, thus the algorithm has the ability to resist plaintext attacks. Theoretical analysis and empirical results show that the algorithm has excellent numerical statistical analysis results, which corroborate that it has good confusion, diffusion and avalanche effects, and is able to resist various common cryptographic attacks. The frequency domain image encryption scheme proposed in this paper is a preferred high-security digital image privacy protection technique, so it has good application prospects.

1. Introduction

With the rapid development of communication technology and network technology, various forms of data and information can be more frequently and widely disseminated through the network, which brings convenience to people's lives while posing new challenges to information security [1–3]. As one of the most intuitive and common data types in information dissemination, images contain a large amount of private information [4–6]. Therefore, considering encryption of images can effectively prevent the leakage of important information during transmission [7–9]. Digital images are characterized by high data redundancy, high correlation, and large amount of data, which makes a large number of pseudo-random numbers are needed as key streams in the process of encrypting images, and this promotes the research of pseudo-random numbers [10–12]. Chaos is widely used and highly respected in the field of image encryption due to its unpredictability, pseudo-randomness, and high sensitivity to initial values. At present, a variety of encryption methods have been proposed, including quantum cipher [13–15], bit-level encryption [16–18], discrete wavelet transform [19–21], thumbnail-preserving encryption [22–24], biological coding [25–27], discrete cosine transform [28–30], Fourier transform [31–33], chaos theory [34–40] and so on [41–45].

An overview of the international situation, many scholars have achieved a series of important theoretical and application results in the use of chaos for image encryption [46–48]. In 2021, [49] proposed a new secure image encryption method based on hyperchaotic system, which uses hardware-efficient reconfigurable PRNG and S-box structure. The results show that the proposed encryption method can be safely used for image encryption. In 2022, [50] proposed a symmetric plaintext-related image cryptosystem based on S-box scrambling framework for the defects of high computational complexity, poor scrambling effect, insufficient correlation between encryption algorithm and plaintext, and low sensitivity of plaintext. A large number of experimental simulations and security analysis show that the proposed cryptosystem has excellent computational efficiency and can effectively resist various existing attacks, including strong chosen-plaintext attacks. In 2023, [51] proposed a strong S-box construction method and a secure image encryption algorithm based on PWQPCM model. In this algorithm, a new method of pixel segmentation encryption, S-box replace and diffusion encryption is combined. It has the advantages of good anti-data loss performance, low time cost, and flexible adjustment of security strength. However, with the increasing complexity of the encryption system, the growth of data volume shows an exponential trend, and it is becoming more and more important to reduce the storage space and transmission bandwidth of data. In the existing chaotic image encryption research, the performance of chaos and algorithm has a great influence on the security and efficiency of the cryptosystem. It is particularly important and urgent to explore an image encryption algorithm that uses chaotic mapping to resist various illegal attacks.

This paper proposes a chaotic image encryption scheme based on three-dimensional S-box and IWT. The experimental results show that the algorithm has excellent encryption effect and good encryption efficiency, and the proposed image encryption algorithm can resist various illegal attacks safely.

The main innovations and contributions of this paper are as follows:

- Many existing encryption schemes are based on time domain encryption, which is not robust enough or does not allow for efficient encryption. This encryption scheme uses IWT to enable high-quality restoration of images, reduce computational complexity, improve real-time performance and efficiency, and save storage space.
- This encryption scheme optimizes the traditional two-dimensional pixel-level S-box to a three-dimensional S-box, so that it has stronger replace ability, which can effectively resist statistical analysis and improve the security of encryption.
- Many of the existing encryption algorithms have unreasonable structures. Without plaintext correlation or ciphertext feedback, it is vulnerable to known plaintext or selected plaintext attacks. This color image encryption algorithm uses plaintext association to generate dynamic chaotic key, which greatly improves the ability to resist cryptographic attacks.
- Many of the existing encryption schemes are based on pixel-level encryption. The encryption granularity is insufficient, and the pixel-level scrambling has security risks. And there is no correlation before and after the encryption process. This color image encryption algorithm uses bit-level permutation and ciphertext cross-diffusion. The experimental results show that the algorithm effectively improves the security.

The rest of this article is organized as follows. Section 2 briefly introduces the chaotic system and wavelet transform algorithm. Section 3 proposes the three-dimensional S-box and image encryption algorithm designed in this paper. In Section 4, the experimental results and simulation results are given. The last part is the conclusion of the paper.

2. Relevant theories

2.1. Chaotic system 2D-SFMH

Chaos comes from the nonlinear dynamic system, and the dynamic system describes any process that changes with time. This process has the characteristics of non-divergence, non-convergence and non-periodicity, and has a very sensitive dependence on the initial value. These characteristics are in line with the requirements of stream ciphers. The mapping used in this paper is a two-dimensional hyperchaotic mapping 2D-SFMH chaotic system [52] coupled by classical sinusoidal mapping and mathematical functions in the model structure. The equation is expressed as:

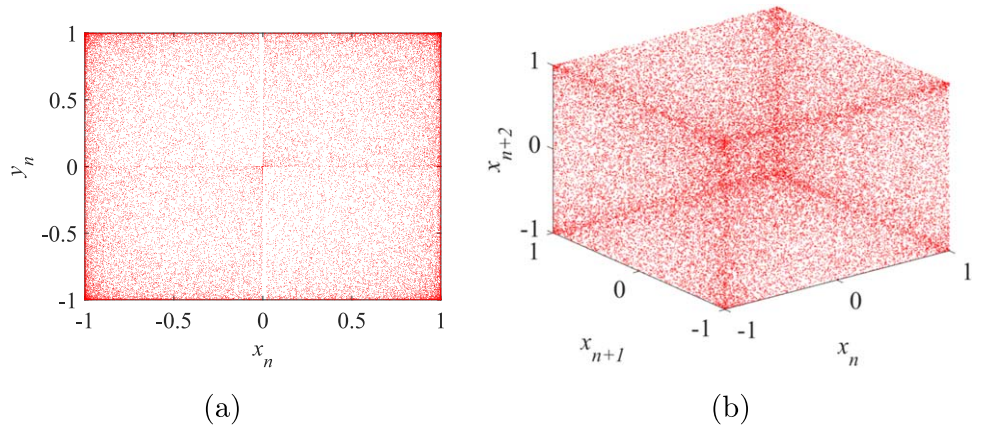

Figure 1. Phase diagram: (a) Chaotic attractor; (b) neighboring variables of 2D-SFHM.

$$\begin{cases} x_{n+1} = \sin\left(\dfrac{\alpha\pi^2}{x_n y_n}\right) \\ y_{n+1} = \sin(b\pi^2 x_n(1 - y_n)) \end{cases} \tag{1}$$

where α and b are recursive parameters, and variables x_{n+1} and y_{n+1} are generated by iterating initial variables x_n and y_n.

2.2. Phase diagram

The trajectory of chaotic attractors can intuitively reflect the behavior of nonlinear systems. Figure 1(a) shows that the trajectory of 2D-SFHM with control parameters is $\alpha = 10$, b = 10, and figure 1(b) represents the relationship between three adjacent variables x_n, x_{n+1}, x_{n+2}.

2.3. Integer wavelet transform

Discrete Wavelet Transform (DWT) has the advantages of multiresolution analysis and spatial frequency localization properties, it can decompose the image into high frequency part and low frequency part, the low frequency component contains most of the energy signals of the image, while the high frequency part represents the detail information. Doing one wavelet transform on the image, four subbands can be obtained *LL*, *LH*, *HL* and *HH*. The discrete wavelet transform has a very important role in the field of signal processing. However, due to the fact that its return value in the wavelet domain is in the form of a floating-point number, which leads to its tendency to cause truncation errors in image encryption and image reconstruction, affecting the recovery effect. The IWT enables the original signal to be mapped from integer to integer, and the Haar Lift Transform is a well-known retrievable transform of this type, whose normal and inverse transforms are as follows:

$$\begin{cases} d_{1,n} = s_{0,2n+1} - s_{0,2n} \\ s_{1,n} = s_{0,2n} + \lfloor d_{1,n}/2 \rfloor \end{cases} \tag{2}$$

$$\begin{cases} s_{0,2n+1} = d_{1,n} + s_{0,2n} \\ s_{0,2n} = s_{1,n} - \lfloor d_{1,n}/2 \rfloor \end{cases} \tag{3}$$

where $s_{i,n}$ and $d_{i,n}$ are the nth low-frequency and higher-frequency coefficients, for layer i of the integer wavelet transform of the image, respectively.

As shown in figure 2, the plain image is decomposed into low frequency coefficients and high frequency coefficients by wavelet transform. *LL* denotes low frequency, *LH*, *HL*, *HH* denotes high frequency.

2.4. The proposed three-dimensional S-box

2.4.1. Constructing a three-dimensional S-box

S-box is a basic structure in the field of cryptography, its function is to realize the nonlinear substitution of data, since S-box is nonlinear, it provides better security compared to other algorithms, and its S-box metrics directly determine the goodness of cryptographic algorithms. In this section, we propose a novel algorithm to construct a three-dimensional S-box based on 2D-SFMH chaotic mapping. The traditional two-dimensional S-box is optimized to three-dimensional S-box, so that it has more powerful substitution ability in the encryption process. The following are the advantages of comparing the two-dimensional S-box:

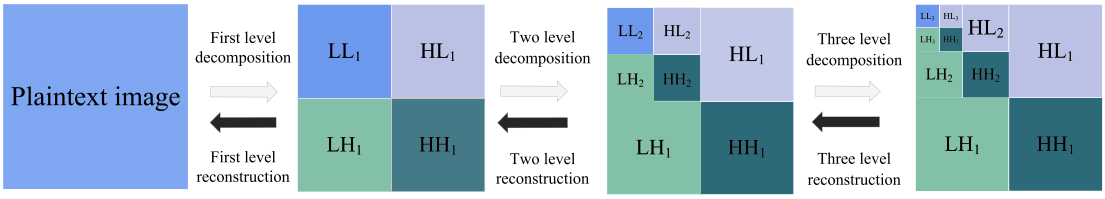

Figure 2. Three-stage IWT flow chart.

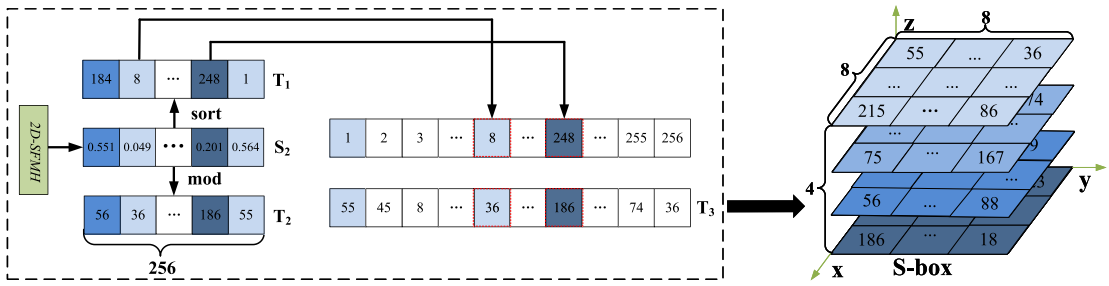

Figure 3. Three-dimensional S-box generation flow chart.

(1) Higher dimensions: Three-dimensional S-box have higher dimensions than two-dimensional substitution box, which means that more S-box data can be provided to obfuscate and displace plaintexts. By operating in three dimensions, chaotic mapping can operate in a larger state space, increasing the complexity and randomness of the system.

(2) Strengthened nonlinearity: Three-dimensional S-box can provide stronger nonlinear properties than two-dimensional S-box. In chaotic systems, nonlinearity is one of the key factors to achieve encryption properties. By introducing additional dimensions, the three-dimensional S-box can better increase the complexity of the data, thereby improving the ability to resist cryptanalysis.

(3) More resistant to attack: The high dimensionality of the three-dimensional S-box increases the difficulty for attackers to perform key space and state space analysis. The state transformation in the three-dimensional S-box is more complex, which makes it more difficult for the attacker to analyze and break the encryption algorithm.

Different from the traditional two-dimensional S-box generation method, this paper generates a three-dimensional S-box model from a pseudo-random sequence. Figure 3 shows the block diagram of S-box generation. Firstly, appropriate initial state values and system parameters are selected to generate S_2 iteratively for the chaotic system, and then two sequences are obtained by modeling and indexing S_2, which are the modeling sequence and the indexing sequence, and then the indexing sequence is used to rearrange the modeling sequence and reconstruct it into a three-dimensional S-box. The detailed generation of $8 \times 8 \times 4$ S-box can be described as follows.

- Step 1: The initial value and the selected control parameters are substituted into the chaotic system to iteratively generate a chaotic sequence, and the first 5000 values are removed.
- Step 2: The chaotic sequence is multiplied by the 10th power of 10 and the floor function is used to convert the fractional value into an integer.
- Step 3: The value between 0 and 255 is obtained by taking the modulus 256 of the sequence.
- Step 4: Now select the first 256 unique values. These values will be in one-dimensional vector T_2.
- Step 5: The chaotic sequence with length of 256 is generated again. By arranging the chaotic sequences, an index vector T_1 can be obtained.
- Step 6: Use the index vector T_1 to reorder the one-dimensional vector T_2 to one-dimensional vector T_3.

Table 1. Reconstructed S-box.

6	156	95	1	111	246	35	93	149	140	145	232	115	179	207	172
165	44	217	114	23	54	65	181	107	9	148	56	206	126	28	116
69	216	113	75	29	74	224	51	178	58	19	166	139	143	32	96
180	219	253	21	110	186	222	80	220	236	184	2	85	84	47	128
234	173	36	57	24	189	235	105	127	225	240	26	202	142	101	204
176	0	13	155	16	66	52	130	102	88	117	3	55	109	169	227
5	231	131	242	138	168	15	160	154	137	238	129	94	42	112	175
164	195	71	218	141	27	78	62	33	64	20,	230	10	213	211	241
67	68	194	72	82	187	210	248	104	34	123	18	17	103	135	196
200	118	59	100	91	83	159	7	86	48	197	22	25	215	79	229
214	153	157	89	183	76	120	182	243	254	144	132	121	147	133	98
226	12	38	228	185	205	146	212	150	41	221	209	11	70	233	249
122	46	49	167	201	8	97	40	81	203	252	171	119	134	237	63
251	50	152	39	162	151	250	108	124	136	30	31	37	90	191	198
188	92	14	170	4	77	99	190	106	192	73	177	255	245	239	125
244	43	45	158	61	53	199	247	223	174	208	60	87	161	163	193

- Step 7: Arrange the one-dimensional vector T_3 into a three-dimensional S-box with a size of $8 \times 8 \times 4$.

2.4.2. Substitution algorithm for three-dimensional S-box

Firstly, the required encrypted image is preprocessed into an 8-bit binary data stream, and then the 8-bit binary data stream is read to $8 \sim 6$ bits, $5 \sim 3$ bits, $2 \sim 1$ bits, and converted to decimal to add one. Finally, the column data, row data, and high data of the three-dimensional S-box generated in section 2.4 are retrieved.

Algorithm 1. The S-box replace algorithm.

Input: Given algorithmically generated S-box and cipher image $\boldsymbol{LF}_{C1}$
Output: Encrypted image $\boldsymbol{LF}_{C2}$

$\boldsymbol{LF}_{C1}$=dec2bin ($\boldsymbol{LF}_{C1}$, 8);
$\boldsymbol{x}$=bin2dec ($\boldsymbol{LF}_{C1}$(:,8:6));
$\boldsymbol{x}=\boldsymbol{x}+1$;
$\boldsymbol{y}$=bin2dec ($\boldsymbol{LF}_{C1}$(:,5:3));
$\boldsymbol{y}=\boldsymbol{y}+1$;
$\boldsymbol{z}$=bin2dec ($\boldsymbol{LF}_{C1}$(:,2:1));
$\boldsymbol{z}=\boldsymbol{z}+1$;
for $i \leftarrow 1$ to HW **do**
$\boldsymbol{LF}_{C2}(i)$ = S-box ($\boldsymbol{x}$(i), $\boldsymbol{y}(i)$, $\boldsymbol{z}(i)$);
a ***end***

2.5. Performance analysis

S-box is the only nonlinear structure of AES. The S-box mainly plays the role of confusion and diffusion in the cryptosystem. In section 2, a hyperchaotic system is introduced to generate a chaotic sequence. The chaotic sequence is used to scramble the number $0 \sim 255$ without repetition, and then the number is rearranged into a $8 \times 8 \times 4$ matrix through section 2.4. In order to facilitate the study of its performance, it is reconstructed into a 16×16 size matrix, and the obtained S-box is shown in table 1. Next, we will study the performance of S-box from six aspects, such as the bijective, nonlinear and strict avalanche criterion of S-box.

2.5.1. Bijectivity

Adamas and Tavares [53] proposed the conclusion that f is bijective if the sum of the linear operations of the Boolean functions f_i of the components of the S-box of $n \times n$ is 2^{n-1}.

$$wt\left(\sum_{i=1}^{n} a_i f_i\right) = 2^{n-1} \tag{4}$$

where $a_i \in \{0, 1\}$ and $a_i (i = 1, 2 \cdots n)$ are not both 0. $wt()$ is the Hamming weight. According to the S-box construction method, the S-box constructed in this paper is bijective.

Table 2. Nonlinearity of S-box.

S-boxes Method	1	2	3	4	5	6	7	8	Avg
The proposed	108	104	106	106	106	96	104	104	104.25

Table 3. The SAC matrix of S-box is proposed.

0.4219	0.4688	0.5469	0.5312	0.5156	0.5000	0.625	0.4531
0.5469	0.5156	0.4219	0.5156	0.4375	0.5625	0.4688	0.4531
0.4688	0.4375	0.5000	0.6094	0.4688	0.5156	0.4219	0.4844
0.5000	0.4688	0.4531	0.5000	0.5469	0.5000	0.5000	0.4844
0.5000	0.5156	0.5156	0.4688	0.4844	0.5000	0.5156	0.5312
0.5000	0.4844	0.4688	0.4375	0.5312	0.4688	0.625	0.4375
0.5469	0.4844	0.5000	0.5469	0.5000	0.4375	0.5000	0.5312
0.5156	0.5625	0.5000	0.5469	0.5469	0.4844	0.5000	0.5312

Table 4. The proposed BIC-SAC matrix of S-boxes.

0	0.5020	0.498	0.4922	0.5098	0.5117	0.5078	0.4727
0.5020	0	0.5156	0.5020	0.502	0.4902	0.5215	0.5293
0.4980	0.5156	0	0.5254	0.5156	0.5078	0.5039	0.5215
0.4922	0.5020	0.5254	0	0.4863	0.5020	0.4844	0.4980
0.5098	0.5020	0.5156	0.4863	0	0.5059	0.4824	0.4746
0.5117	0.4902	0.5078	0.5020	0.5059	0	0.4961	0.5098
0.5078	0.5215	0.5039	0.4844	0.4824	0.4961	0	0.5195
0.4727	0.5293	0.5215	0.4980	0.4746	0.5098	0.5195	0

2.5.2. Nonlinearity

Nonlinearity [54] is a test of a cryptographic function's ability to resist linear attacks. The ability of a function to resist linear attacks is directly proportional to its nonlinearity. The nonlinearity of an n-bit Boolean function is defined as follows:

$$N_f = 2^{n-1} - \frac{1}{2} \max_{w \in GF(2^n)} |S_f(w)| \tag{5}$$

where $S_f(w)$ is the Walsh cycle spectrum of $f(x)$. The proposed algorithm can construct the nonlinearity of S-box. As shown in table 2, it can be seen that the proposed S-box has high nonlinearity.

2.5.3. Strict avalanche criterion (SAC)

The strict avalanche effect is introduced to observe the intuitive nonlinear characteristics. The strict avalanche criterion [55] means that when one input of the Boolean function is changed, half of the output value will change, that is, the change probability of each output bit is 0.5. The independent matrix is used to obtain the SAC value of the S-box. If the S-box satisfies SAC, each element of the independent matrix is close to 0.5. Table 3 shows the matrix. We can see from table 3 that the value of each element is close to 0.5.

2.5.4. Output bit independence criterion (BIC)

Bit independence criterion (BIC) [56] is another ideal feature of encryption algorithms. For S-box: $\{0, 1\}^n \rightarrow \{0, 1\}^n$, it can satisfy BIC when for all $i, j, k \in (1, 2, \cdots, n)$ and $j \neq k$, the change of input bit i can cause the independent change of output bit j and k. Table 4 and table 5 show the BIC-SAC and BIC-nonlinearity test results of the S-box. It can be seen that the minimum and maximum values of BIC-SAC are 0.4727 and 0.5254, respectively, and the mean value is 0.5, which is equal to the ideal value. The minimum value of BIC-nonlinearity is 96, the maximum value is 108, and the average value is 103.42, which indicates that S-box exhibits good performance under this criterion.

2.5.5. Difference approximation probability (DP)

The difference approximation probability [57] DP_f represents the XOR distribution of the input and output of the Boolean function. Given an input difference $\triangle x$, the probability that the output is $\triangle y$ is the highest. The

Table 5. The BIC-Nonlinearity matrix of S-box is proposed.

0	102	104	102	102	108	106	100
102	0	104	106	102	104	104	108
104	104	0	100	104	104	104	104
102	106	100	0	108	102	102	104
102	102	104	108	0	100	96	104
108	104	104	102	100	0	106	100
106	104	104	102	96	106	0	106
100	108	104	104	104	100	106	0

Table 6. Comparison of S-box performance.

S-boxes	Nonlinearity	SAC	BIC-SAC	BIC-NL	Dp Max	Lp Max
Method	Avg	Avg	Avg	Avg		
[59]	103	0.5039	0.5030	100.35	0.500	0.1484
[60]	104.75	0.5041	0.5050	104	0.0390	0.1406
The proposed	104.25	0.5089	0.5031	103.42	0.0391	0.1406

smaller DP_f is, the stronger its resistance to differential attacks is. The maximum value of S-box DP proposed in this paper is 0.0391.

2.5.6. Linear approximation probability (LP)

The linear approximation probability [58] means that when two masks and are arbitrarily selected, the mask C_x operation is performed on all possible values of the input value x, and the mask C_y operation is performed on the output value $S(x)$ of the corresponding S-box. The maximum number of the same results obtained by the operation of the input value and the output mask is the maximum linear approximation. The maximum value of S-box LP constructed in this paper is 0.1406.

2.5.7. Performance comparisons

As shown in table 6, the constructed S-box has stronger encryption characteristics than some other methods, which lays some foundation for subsequent research.

3. Proposed image encryption system

3.1. Encryption algorithm

The total framework of the encryption system proposed in this paper is shown in figure 4. Firstly, the dynamic key with plaintext correlation is generated by plaintext hash value, and the chaotic pseudo-random sequence is generated and obtained by using the key, then, the low-frequency image and the high-frequency image are obtained by IWT transformation of the plaintext image, since the low-frequency component contains most of the energy signal of the image, and the high-frequency part represents the detail information, the low-frequency image rich in information is weighted and encrypted. The steps are: bit-level permutation, S-box replace, and ciphertext cross-diffusion, while the high-frequency image with less information is only encrypted by lightweight scrambling. The specific steps of the encryption algorithm is as follows:

Step 1: Plaintext association key and IWT processing plaintext

Firstly, the plaintext image is input and the hash table is used to obtain the SHA-256 hash value of the plaintext image, which is encoded as the initial value Key that conforms to the chaotic initial value interval. Then the image is decomposed into *LL*, *LH*, *HL*, *HH* four subbands by IWT, and then the haar filter is used to calculate the coefficients of each sub-band, and the image matrix is mapped from the spatial domain to the frequency domain. Finally, the mapped data is processed into a pixel value range of $0 \sim 255$, and a low-frequency image LF_C and three high-frequency images HF_C can be obtained. The equation for generating the initial value Key is as follows:

$$\begin{cases} h_i = hex2dec(h) \\ lenh_i = strlength(num2str(h_i)) \\ key_i = h_i/(lenh_i \times 10^{10}) \end{cases} \tag{6}$$

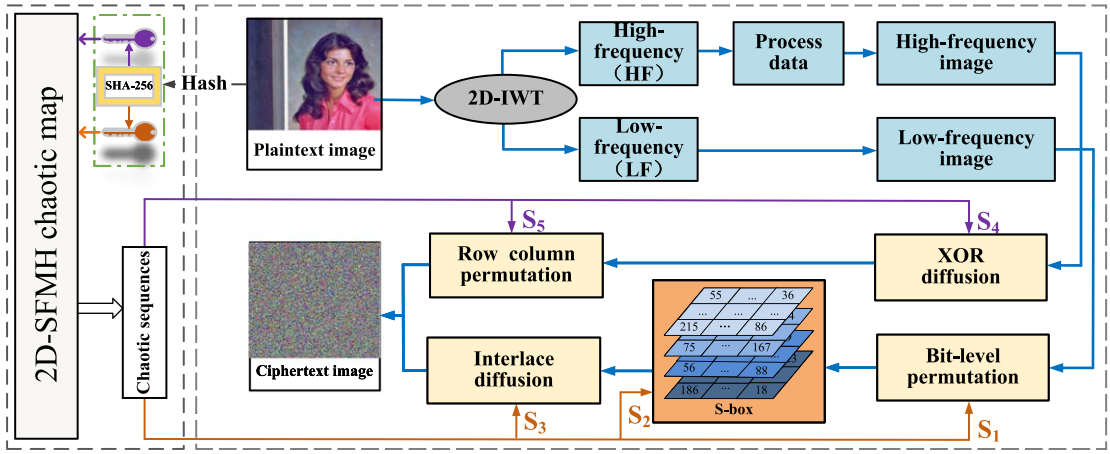

Figure 4. The overall block diagram of image encryption.

Table 7. Comparison of adjacent pixel correlation between some plaintext images and their ciphertext images.

Image name	Algorithm	Horizontal	Vertical	Diagonal	Back-diagonal
	this paper	−0.0007	−0.0285	−0.0023	0.0112
5.1.12	[49]	0.0055	−0.0049	0.0001	/
	[51]	0.0260	0.0039	0.0027	/
	this paper	0.0083	−0.0247	0.0059	−0.0118
5.2.08	[49]	0.0041	0.0014	0.0029	/
	[51]	−0.0026	0.0021	−0.0024	/
	this paper	0.0334	−0.0105	0.0255	0.0257
5.3.02	[49]	−0.0376	−0.0473	−0.0301	/
	[51]	−0.0025	−0.0002	−0.0009	/
	this paper	−0.0159	−0.0075	0.0025	−0.0165
7.1.03	[49]	0.0001	0.0002	0.0031	/
	[51]	−0.0034	−0.0004	0.0017	/

where h is the hexadecimal hash value, h_i is the decimal hash value, and $lenh_i$ is the hash length. key_i is the initial value required for low-frequency encryption and high-frequency encryption.

Step 2: Pseudo-random sequence preprocessing

The key of this algorithm includes the initial value Key and control parameters of 2D-SFMH chaos. The initial value obtained by Step 1 and the selected control parameters are substituted into the chaotic system to iteratively generate chaotic sequences and process them into the sequences required for encryption. The specific processing equation is as follows:

$$\begin{cases} S_1 = (\lfloor |X_1| \rfloor \times 10^{10}) \bmod 256 \\ S_2 = (\lfloor |X_2| \rfloor \times 10^{10}) \bmod 256 \\ S_3 = (\lfloor |X_3| \rfloor \times 10^{10}) \bmod 256 \\ S_4 = (\lfloor |X_4| \rfloor \times 10^{10}) \bmod 256 \\ S_5 = (\lfloor |X_5| \rfloor \times 10^{10}) \bmod 256 \end{cases} \tag{7}$$

where $\lfloor \cdot \rfloor$ denotes the downward rounding. S_1, S_2, S_3, S_4, S_5 are processed random sequences. $|\cdot|$ denotes the absolute value. X_i is denoted as the chaotic sequence.

Step 3: Bit permutation

Firstly, the low-frequency image with the size of $H \times W$ obtained by Step 1 is converted into an 8-bit binary data stream, and then the binary image with the size of $8H \times W$ is reconstructed. The pseudo-random sequence S_1 generated by Step 2 is used for row and column permutation, and it is converted into decimal. Finally, it is reconstructed into $H \times W$. The specific operation is as follows:

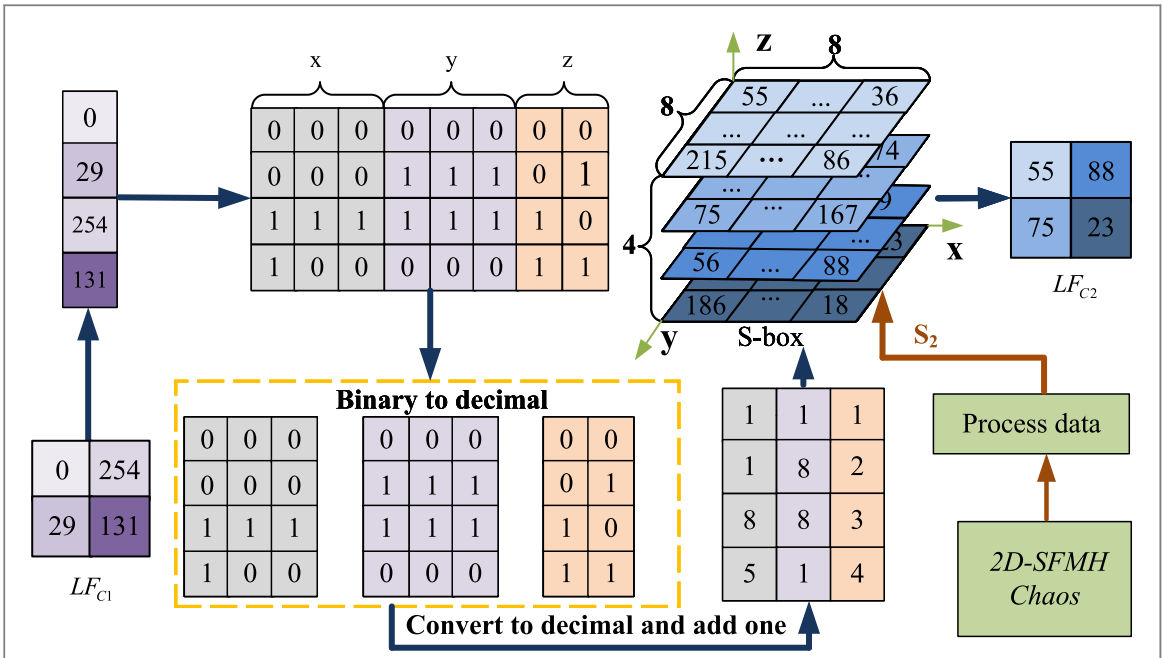

Figure 5. Three-dimensional S-box replace flowchart for images of size 2 × 2.

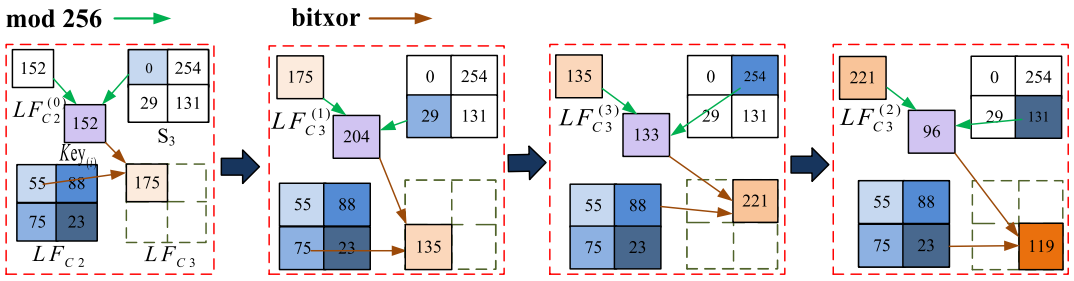

Figure 6. Ciphertext interleaving diffusion encryption schematic diagram.

$$\begin{cases} [\sim, indexH] = sort(S_1(1{:}\ 8H)) \\ [\sim, indexW] = sort(S_1(8H+1,\ W)) \\ LF_{C1}(i,j) = LF_C(indexH(i),\ indexW(j)) \end{cases} \tag{8}$$

where LF_C represents the image after bit-level permutation, i, j indicates the binary image size, $i = 1, 2, ..., 8H$, $j = 1, 2, ..., W$.

Step 4: S-box replace

Firstly, the S-box is constructed by using the pseudo-random sequence S_2 generated by Step 2 and the algorithm in section 2.4. Then, the intermediate ciphertext LF_{C1} after bit substitution is converted into a one-dimensional pixel sequence and converted into a binary number. The encrypted intermediate ciphertext image LF_{C2} is retrieved by section 2.4 algorithm 1. The specific encryption steps are shown in figure 5.

Step 5: Ciphertext interleaved diffusion

Using the S_3 sequence, ciphertext interleaved diffusion [61] is performed on the intermediate ciphertext LF_{C2} to obtain the ciphertext image LF_{C4} in the low-frequency portion. The encryption method divides the image into two sub-blocks, and after one round of pixel encryption of the sub-blocks in a parallel manner, the generated ciphertext is encrypted in two rounds to achieve the ciphertext interleaved diffusion technique, in which one round of micro-encryption process is shown in figure 6.

From the above steps, the intermediate ciphertext matrix LF_{C2} is obtained, and its mathematical expression is as follows:

$$LF_{C2} = \begin{pmatrix} LF_{C2}^{1} & LF_{C2}^{2} & \cdots & LF_{C2}^{N} \\ \vdots & \vdots & \ddots & \vdots \\ LF_{C2}^{(M-1)+1} & \cdots & \cdots & LF_{C2}^{L} \end{pmatrix} \tag{9}$$

where the pixel size is M rows and N columns and the total number of pixels is $L = M \times N$.

Next, interleaved diffusion is performed for the LF_{C2}. The first half of the sub-block consists sequentially of the pixel sequence $\{LF_{C2}^{1}, LF_{C2}^{2}, \ldots, LF_{C2}^{(L/2)}\}$ and the second half of the sub-block consists sequentially of the pixel sequence $\{LF_{C2}^{L/2+1}, LF_{C2}^{L/2+2}, \ldots, LF_{C2}^{(L)}\}$.

- Preprocessing the first bit of the encrypted sub-block.

When $i = 1$, the first pixel of the first half sub-block is encrypted. The encryption is as follows:

$$\begin{cases} Key_{(i)} = mod(LF_{C2}^{(0)} + S_3^{(i)}, 256) \\ LF_{C3}{}^{(i)} = bitxor(LF_{C2}^{(i)}, Key_{(i)}) \end{cases} \tag{10}$$

At the same time, the first pixel of the second half sub-block is encrypted. The encryption equation is as follows:

$$\begin{cases} Key_{(L/2+i)} = mod(LF_{C3}{}^{(i)} + S_3^{(L/2+i)}, 256) \\ LF_{C3}{}^{(L/2+i)} = bitxor(LF_{C3}{}^{(L/2+i)}, Key_{(L/2+i)}) \end{cases} \tag{11}$$

where $Key_{(i)}$ is a new sequence generated from $LF_{C2}{}^{(i)}$ and $S_3^{(i)}$ by data processing, $S_3^{(i)}$ is a pseudo-random sequence generated by Step 2, and $LF_{C2}{}^{(i)}$ is a predefined positive integer in the range of 1 to 255. $LF_{C2}{}^{(i)}$ and $LF_{C3}{}^{(i)}$ are the values of the i th pixel of the original image and the encrypted image, respectively.

- Interleaved diffusion for sub-blocks.

The arithmetic progression of i is increased to a tolerance of 1, and at the same time, the i th pixel of the previous sub-block is encrypted. The encryption is as follows:

$$\begin{cases} Key_{(i)} = mod(LF_{C3}^{(L/2+i-1)} + S_3^{(i)}, 256) \\ LF_{C3}{}^{(i)} = bitxor(LF_{C2}^{(i)}, Key_{(i)}) \end{cases} \tag{12}$$

The encryption operation is performed on the i th pixel of the second half sub-block as follows:

$$\begin{cases} Key_{(L/2+i)} = mod(LF_{C3}{}^{(i)} + S_3^{(L/2+i)}, 256) \\ LF_{C3}{}^{(L/2+i)} = bitxor(LF_{C3}{}^{(L/2+i)}, Key_{(L/2+i)}) \end{cases} \tag{13}$$

where $Key_{(i)}$ is a new sequence generated by $LF_{C2}{}^{(i)}$ and $S_3^{(i)}$ through data processing, $S_3^{(i)}$ is a pseudo-random sequence generated by Step 2. $LF_{C3}{}^{(i)}$ is the value of the i th pixel of the encrypted image.

- Second round of interlaced diffusion.

The intermediate ciphertext LF_{C3} replace the intermediate ciphertext LF_{C2}, $LF_{C3}{}^{(L)}$ replace $LF_{C2}{}^{(0)}$, and the first round of encryption operation is repeated to obtain the ciphertext image LF_{C4} of the low frequency part.

Step 6: High frequency image processing

Using the pseudo-random sequence S_4, S_5 of Step 2, the three high-frequency images with a size of $H \times W$ obtained in the first step are subjected to row-column permutation and XOR operation in turn to obtain the encrypted high-frequency image. The specific operation is as follows:

$$\begin{cases} [\sim, indexH] = sort(S_4(1{:}\ H)) \\ [\sim, indexW] = sort(S_4(H+1,\ W)) \\ HF_{C1}(i, j) = HF_C(indexH(i),\ indexW(j)) \\ HF_{C2}(i, j) = HF_{C1}(i, j) \otimes S_5(i, j) \end{cases} \tag{14}$$

where HF_{C1} represents the image after bit-level permutation, HF_{C2} represents the image after XOR.

Step 7: Generate ciphertext

The low-frequency image and the high-frequency image generated above are merged to obtain the final ciphertext image C.

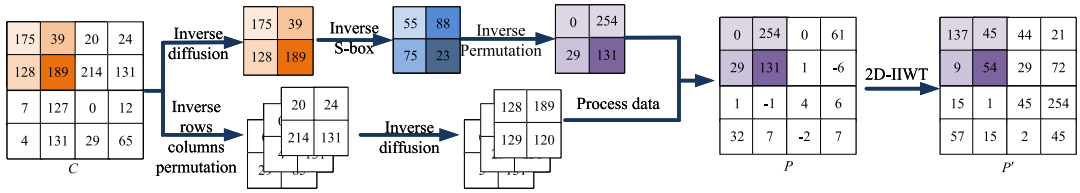

Figure 7. Decryption diagram of image with size of 4 × 4.

3.2. Decryption process

Decryption is the inverse process of encryption. Before decrypting the image, the key used in the encryption process needs to be transmitted to the decryption end through a secure channel. In the decryption process, the ciphertext image is first decomposed into low-frequency and high-frequency parts by IWT. Then, the low-frequency part performs reverse ciphertext interlaced diffusion, S-box replace and inverse bit permutation, and then performs reverse row-column permutation and XOR operation on the high-frequency part. Finally, the inverse IWT is performed on the transformed high-frequency and low-frequency data to obtain the final plaintext image. The image decryption process is shown in figure 7.

4. Experimental results and analysis discussion

4.1. Experimental environment

For the experimental platform, we used a personal computer (PC) host with MATLAB R2022a experimental software installed. The processor of the PC is 11th Gen Intel(R) Core(TM) i5-11400H CPU with 2.70 GHz, the memory size is 16 GB, the hard disk size is 1 TB, and the operating system is Windows 10. The image data selected for the experiment is also USC-SIPI.

4.2. Experimental results and analysis

4.2.1. Histogram analysis

The histogram intuitively depicts the distribution of gray levels in the image and their respective frequencies. In general, the histogram of the plain image shows a certain statistical pattern, and the statistical characteristics of the encrypted image histogram show a noise-like distribution, which masks the main information of the image, thereby improving the ability to resist statistical analysis attacks. We select five plaintext images with different sizes and the corresponding histograms, and obtain the corresponding ciphertext and histograms by encryption algorithm, as shown in figure 8.

4.2.2. Correlation analysis of adjacent pixels

Usually, the plaintext image contains a large number of pixels with high neighborhood correlation, and the ciphertext image encrypted by a good encryption algorithm will not associate any pixel and its adjacent pixels. Therefore, the secure encryption scheme aims to generate ciphertext images, in which adjacent pixels exhibit negligible correlation. In order to calculate and compare the correlation between adjacent pixels in plaintext and ciphertext images, we use the following steps. First, we randomly select 3000 pairs of adjacent pixels from the plaintext image and the ciphertext image, and then calculate the correlation coefficients of the adjacent pixels in the horizontal, vertical, diagonal, and back-diagonal directions according to equation (13). The correlation coefficient is calculated as follows:

$$r_{xy} = \frac{\sum_{i=1}^{M}\left(x_i - \frac{1}{M}\sum_{j=1}^{M}x_j\right)\left(y_i - \frac{1}{M}\sum_{j=1}^{M}y_j\right)}{\sqrt{\sum_{i=1}^{M}\left(x_i - \frac{1}{M}\sum_{j=1}^{M}x_j\right)^2}\sqrt{\sum_{i=1}^{M}\left(y_i - \frac{1}{M}\sum_{j=1}^{M}y_j\right)^2}} \tag{15}$$

where x_i and y_i constitute the first pair of horizontal/vertical/diagonal/anti-angle adjacent pixels, M is the total number of horizontal/vertical/diagonal/anti-angle adjacent pixels. The adjacent pixel correlation data of the encrypted image is shown in figure 9. Table 7 shows the correlation of two horizontal, vertical, diagonal and anti-diagonal adjacent pixels in the plaintext image and ciphertext image of 'House.tiff', respectively. It can be seen from the experimental data that the correlation coefficient of the ordinary image is close to 1, and the correlation coefficient of the encrypted image is approximately equal to 0. This shows that the proposed encryption scheme generates images with negligible correlation between adjacent pixels. Therefore, the scheme proposed in this paper is secure against statistical attacks.

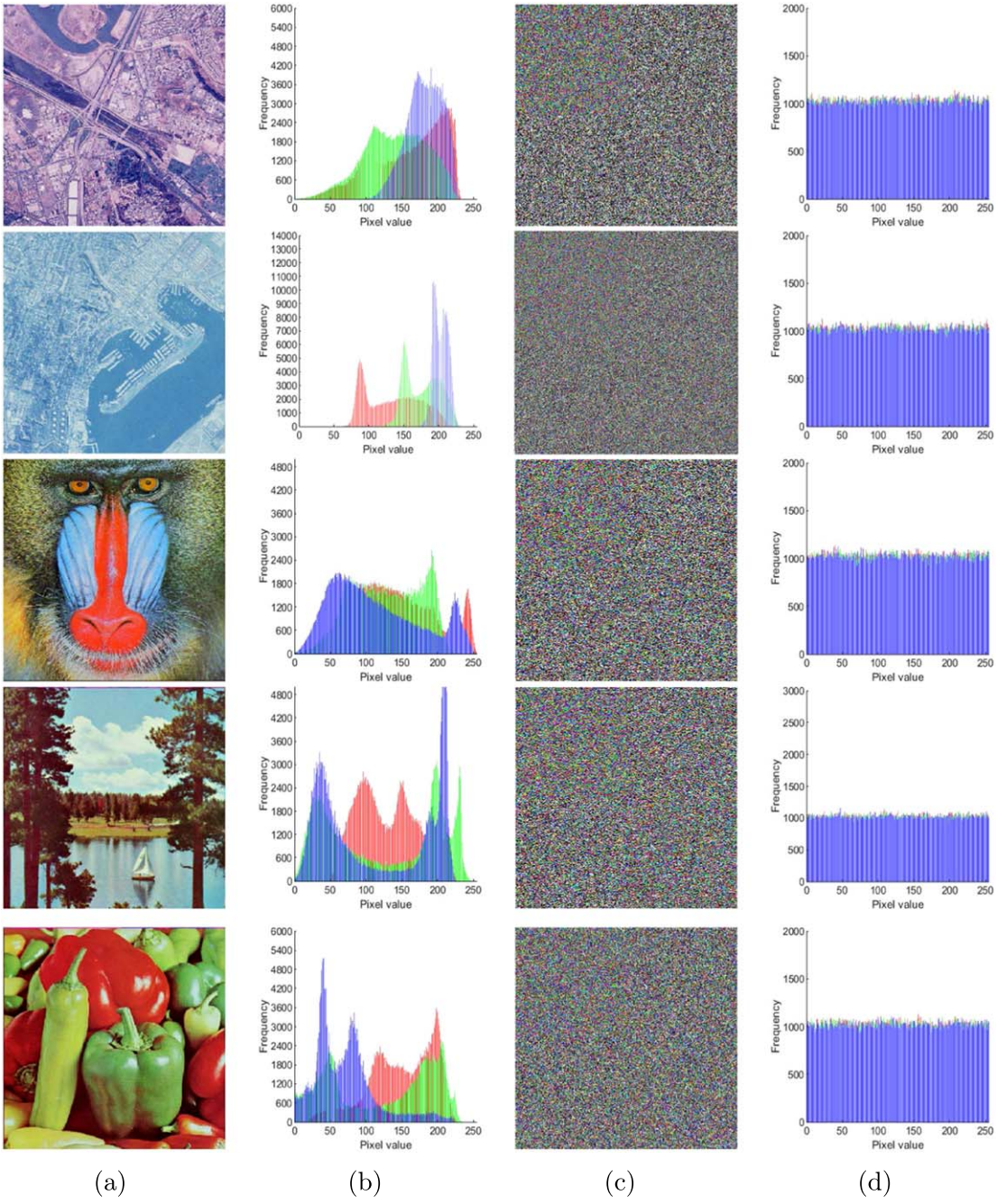

Figure 8. Images before and after encryption: (a) plaintext image; (b) histogram of plaintext image; (c) ciphertext image; (d) histogram of ciphertext image.

4.2.3. Differential statistics

The difference between the two images can be quantified using two criteria: Number of Pixels Change Rate (NPCR) and Unified Average Changed Intensity (UACI). In differential attacks, attackers often make slight changes to the plaintext image, using a specific algorithm to encrypt it before and after the adjustments, aiming to reveal their relationship. NPCR and UACI are described as follows:

$$\begin{cases} \mathrm{NPCR} = \dfrac{1}{H \times W} \times \sum_{i=1}^{H} \sum_{j=1}^{W} D(i,j) \times 100\% \\ \mathrm{UACI} = \dfrac{1}{H \times W} \times \sum_{i=1}^{H} \sum_{j=1}^{W} \dfrac{|v_1(i,j) - v_2(i,j)|}{255} \times 100\% \end{cases} \tag{16}$$

where $H \times W$ is the size of the image. v_1, v_2 respectively for plaintext image change a pixel before and after the ciphertext image. D can be defined by:

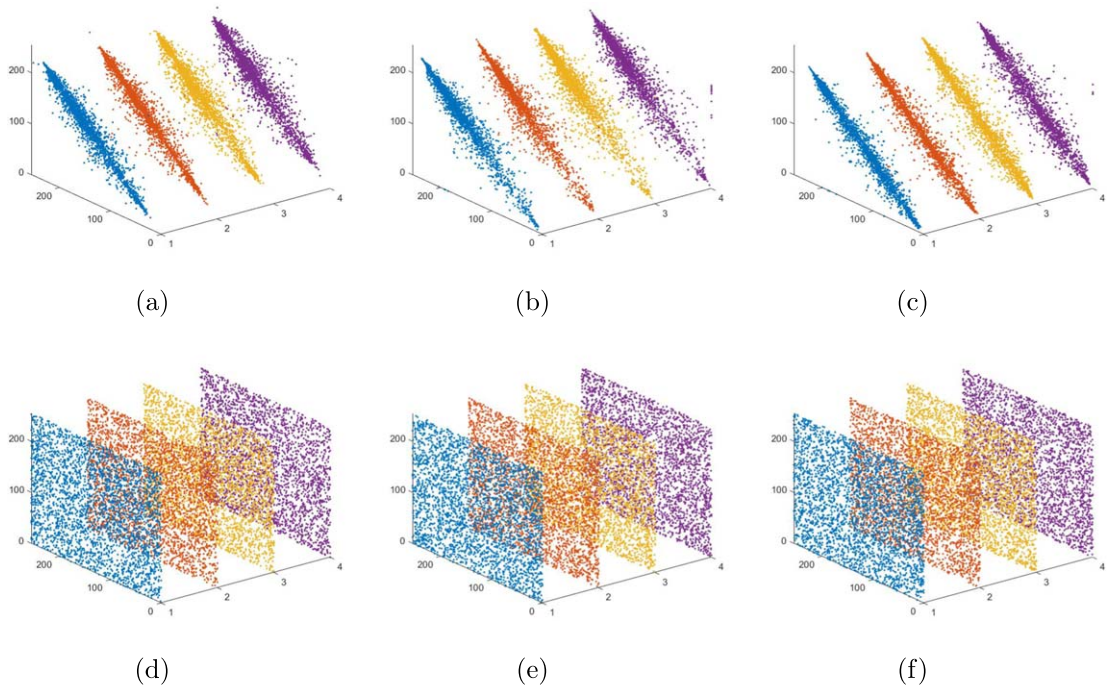

Figure 9. Neighboring Pixel correlation for plaintext and ciphertext images: (a) plaintext R channel; (b) plaintext G channel; (c) plaintext B channel; (d) ciphertext R channel; (e) ciphertext G channel; (f) ciphertext B channel.

Table 8. NPCR and UACI values.

Picture	NPCR	UACI
2.1.02.	99.6021	33.4181
2.1.05.	99.6174	33.4099
2.1.06.	99.6105	33.3263
4.1.06.	99.6002	33.4181
5.1.10.	99.6227	33.5582

$$D = \begin{cases} 0 & v_1(i,j) = v_2(i,j) \\ 1 & v_1(i,j) \neq v_2(i,j) \end{cases} \tag{17}$$

Table 8 shows the algorithm results calculated according to equation (16). we found that table 8 shows the NPCR and UACI values of images with different sizes encrypted by the algorithm. Based on the experimental results, it is evident that the encryption scheme exhibits high sensitivity to alterations in the plaintext image and can effectively withstand chosen plaintext attacks.

4.3. Information entropy analysis

Information entropy is a crucial indicator for assessing the distribution of image gray values and measuring the randomness of image information. It can be defined as follows:

$$H(m) = -\sum_{i=1}^{L} P(n_i)\log_2 P(n_i) \tag{18}$$

where m is the total number of symbol $m(i) \in m$ and $p(n_i)$ denotes the probability of a symbol.

Assume that the information source sends 256 symbols, and we can obtain the theoretical value $H(m)$ by equation (18). The closer to 8, the less likely the attacker will decode the password image. Table 9 shows the comparison of information entropy. It can be seen from table 9 that the experimental results are close to 8, so the proposed algorithm has good information entropy characteristics.

4.4. Image quality analysis

In the field of image processing, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are commonly used as a tool to weigh the quality of encryption. Mean square error (MSE) is a part of PSNR, as

Table 9. Image information entropy.

Image	Size	Proposed Plain image	This paper	[31]	[62] Cipher image	[47]
5.1.10	256 × 256	7.3118	7.9970	7.9966	7.9971	7.9968
5.1.11	256 × 256	6.4523	7.9961	7.9971	7.9699	7.9971
5.1.12	256 × 256	6.7057	7.9967	7.9972	7.9975	7.9973
5.1.13	256 × 256	1.5483	7.9937	7.9970	7.9973	7.9968
5.1.14	256 × 256	7.3424	7.9964	7.9971	7.9967	7.9969
5.2.08	512 × 512	7.2010	7.9992	7.9993	7.9993	7.9992
5.2.09	512 × 512	6.9940	7.9993	7.9993	7.9993	7.9994
5.2.10	512 × 512	5.7056	7.9991	7.9992	7.9992	7.9993
7.1.01	512 × 512	6.0274	7.9992	7.9993	7.9992	7.9993
7.1.02	512 × 512	4.0045	7.9990	7.9993	7.9993	7.9994
7.1.03	512 × 512	5.4957	7.9992	7.9993	7.9992	7.9994
7.1.04	512 × 512	6.1074	7.9992	7.9993	7.9992	7.9993
7.1.05	512 × 512	6.5632	7.9994	7.9993	7.9993	7.9993
7.1.06	512 × 512	6.6953	7.9993	7.9994	7.9992	7.9991
7.1.07	512 × 512	5.9916	7.9992	7.9991	7.9993	7.9992
7.1.08	512 × 512	5.0534	7.9992	7.9993	7.9992	7.9992
7.1.09	512 × 512	6.1898	7.9993	7.9992	7.9993	7.9992

Table 10. PSNR, MSE and SSIM values.

Picture	PSNR	MSE	SSIM
2.1.02	47.2594	0.4151	0.9995
2.1.05	47.2420	0.4178	0.9989
2.1.06	51.8592	0.4238	0.9982
4.1.04	44.2198	0.5126	0.9980
5.1.12	51.8500	0.4246	0.9995

shown in table 10, which is defined as:

$$\begin{cases} \mathrm{MSE} = \dfrac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}(I(i,j) - C(i,j))^2 \\ \mathrm{PSNR} = 10\lg(\dfrac{MAX_I{}^2}{MSE}) \end{cases} \tag{19}$$

where MSE represents the mean square error of plaintext image X and ciphertext image Y. The height and width of the image are represented by H and W respectively, and Q represents the pixel level of the image. SSIM is an index to measure the similarity of two images, as shown in table 10, which is defined as:

$$\mathrm{SSIM(X,\ Y)} = \frac{[2\mu_X\mu_Y + (0.01L)^2][2\sigma_{XY} + (0.03L)^2}{[\mu_X^2 + \mu_Y^2 + (0.01L)^2[\sigma_X^2 + \sigma_Y^2 + (0.03L)^2} \tag{20}$$

where μ_X, μ_Y represents the mean value of image X and Y respectively, represents the standard deviation of image X and Y respectively, and L represents the dynamic range of pixel value. The values of PSNR and SSIM are calculated by using equations (19) and (20) as shown in table 10. The PSNR value of an encrypted image should be about 30 dB, and the range of SSIM should be -1 to 1. The closer the image is, the closer the absolute value of SSIM is to 1. Therefore, the value of SSIM should fluctuate around 0 after encryption.

4.5. Key space

The key space refers to the set of all possible keys that can be used to generate the key. The size of the key space depends on the length of the security key, which is one of the most important characteristics that determine the strength of the cryptosystem. The image encryption algorithm designed in this paper uses a two-dimensional discrete chaotic system, and its key space can be expressed as $S \in \{x, y, r, SHA - 256\}$, where x, y, r is the key parameter with an accuracy of 10^{-16}, and $SHA - 256$ is the hash value introduced to enhance the key space, which can generate 256 bit hash, using the first 128 bits. After calculation, the key space size of the encryption scheme is about $10^{3\times16} \times 2^{128} \approx 2^{287}$, and the key length reaches 287 bit. Therefore, the encryption algorithm in this paper can resist any form of violent attacks. The key space comparison is shown in table 11.

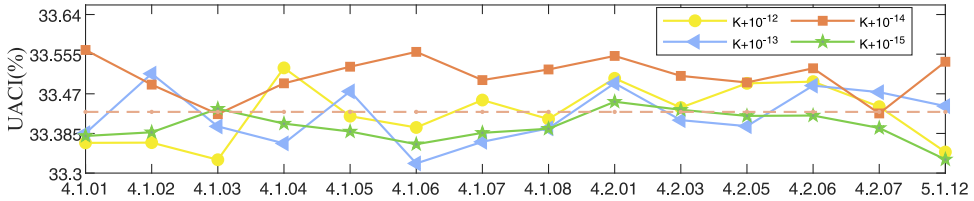

Figure 10. Key sensitivity test UACI results.

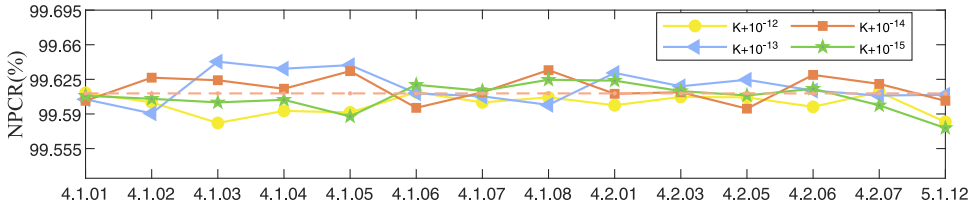

Figure 11. Key sensitivity test NPCR results.

Table 11. Key space.

This paper	[63]	[64]	[62]	[42]
287	154	166	224	234

4.6. Sensitivity analysis

In this section, we assess the algorithm's performance by examining its sensitivity to changes in both the encryption key and plaintext image. This emphasizes the strict requirements of security algorithms for high sensitivity. Even minor alterations to the key or the information within the plaintext image can have a substantial impact on the final encryption outcomes during the process of encryption or decryption. This shows that the performance of the security algorithm not only focuses on the improvement of the encryption intensity, but also on the sensitivity of the input parameters. We use NPCR and UACI to judge the difference between the images generated by the original algorithm and the scrambled algorithm, so as to verify the high sensitivity of the algorithm.

4.6.1. Key sensitivity analysis

The key sensitivity is to analyze the ciphertext obtained when encrypting the same image by using two slightly different keys. In this section, we compare the differences between the obtained ciphertexts by using the correct key and slightly changed key (add 10^{-12}, 10^{-13}, 10^{-14} and 10^{-15}) for encryption. The difference between them was derived by calculating NPCR and UACI, where NPCR and UACI were calculated as shown in equation (16). The results are shown in figure 10 and figure 11. We can find that when the perturbation is added to the key, the average values of NPCR and UACI are 99.6108% and 33.46%, which indicates that the difference between the two ciphertext images is very large.

4.6.2. Plaintext sensitivity analysis

The plaintext sensitivity refers to the degree of change of the corresponding ciphertext when the pixel of the plaintext is changed. If the algorithm is not sensitive to plaintext, attackers are likely to break the algorithm by analyzing the difference between plaintext and ciphertext pairs. Therefore, the plaintext sensitivity of the algorithm is the key to its resistance to plaintext attacks. In this section, we analyze the sensitivity of the proposed algorithm to ordinary images by adding 1 to the pixel values of ordinary images at $(H/3, W/3)$, $(2 \times H/3, W/3)$, $(H/3, 2 \times W/3)$ and $(2 \times H/3, 2 \times W/3)$ to calculate NPCR and UACI. The results are shown in figure 12 and figure 13. From the figure, it can be seen that when the variation of the pixel values at the selected locations is 1, the average NPCR and UACI between their corresponding ciphertexts and the original ciphertexts reaches 99.6046% and 33.43% respectively, which is very close to the desirable values of 99.6094% and 33.46%. This indicates that the cryptographic image has changed significantly. This makes it impossible for an attacker to

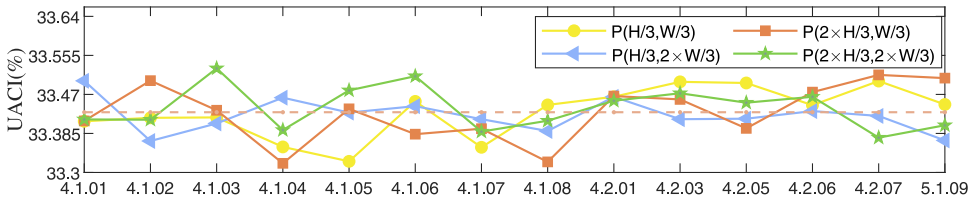

Figure 12. Plaintext sensitivity test UACI results.

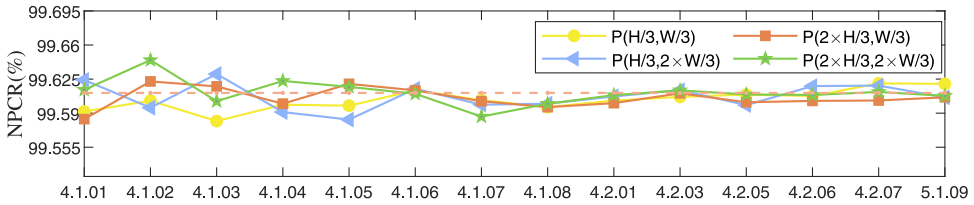

Figure 13. Plaintext sensitivity test NPCR results.

compromise the algorithm by comparing the differences between the ciphertexts and hence the proposed algorithm is sufficient to resist plaintext attacks.

4.7. Chaotic sequence test-NIST test

National Institute of Standards and Technology (NIST) SP800-22 is a quantitative metric for evaluating whether a serial is randomly distributed by performing 15 sub-tests to measure the random distribution of the map. The experiment tests 120 sets of binary hyperchaotic sequences of length 10^6 generated by 2D-SFHM, and the $P-value$ and *Proportion* generated during the process are the critical indicators used to evaluate the random distribution. At the level of α=0.01, $1 > P-value > 0.01$ and $Proportion > 96.28\%$ mean that the sequences are highly random. Table 12 clearly shows that the 15 sub-test results of the generated sequences, which fully satisfy the random distribution requirement.

4.8. Security analysis based on password attack

In the field of cryptography, the famous Kerckhoffs principle holds that the encryption algorithm of a secure cryptosystem should be known publicly by the attacker, and only the key is unknown. The four commonly used cryptanalysis methods from weak to strong are: ciphertext-only attack, known-plaintext attack, chosen-plaintext attack selection, ciphertext attack. However, chosen-plaintext attack and chosen-ciphertext attack are currently considered to be the most effective password attack methods. The main concepts behind these attacks involve carefully selecting specific attack images, such as all-black or all-white images, and then applying algebraic analysis to derive the equivalent key of the original cryptosystem. In this paragraph, we decipher the algorithm by selecting the plaintext attack method. Using comparable attack strategies, we evaluate the security of the image encryption algorithm proposed in this paper. As shown below, figures 14(a)–(d) is specified as the plaintext attack image, and the corresponding ciphertext image is presented in figures 14(e)–(h). As shown in figures 14(i)–(l), their histogram features show noise-like patterns, which are significantly different from figures 14(a)–(d). In order to ensure versatility, we also select a specific plaintext image 14(m)–(p), whose corresponding ciphertext image is displayed in figures 14(q)–(t). In addition, in figures 14(u)–(x), the histograms of these ciphertexts show noise patterns and are significantly different from the original text. Therefore, this method makes it challenging for attackers to carry out penetration attacks. Similarly, it is also difficult to implement selective ciphertext attacks. The main source of these challenges comes from the combination of the substitution-permutation-diffusion structure and the plain text association mechanism. At the same time, the use of substitution-permutation-diffusion structure effectively enhances the avalanche effect and security of the algorithm.

4.9. Performance comparison of multi-round Integer Wavelet Transform to encrypt images of different sizes

When the size of the larger image, we will use multiple rounds of IWT before the above encryption operation. The following figure 15 shows the processing time of images of different sizes after different times of IWT,

Table 12. NIST SP800-22 test of 2D-SFHM.

NO.	Sub-tests	P-value >= 0.0001	Pass rate >= 0.9628
01	Frequency	0.3115	0.9917
02	Block Frequency	0.0106	0.9750
03	Cumulative Sum(F)	0.4528	0.9917
	Cumulative Sum(R)	0.6371	0.9917
04	Runs	0.9320	0.9917
05	Longest runs	0.1866	0.9750
06	Rank	0.7399	0.9917
07	FFT	0.9915	1.0000
08	Non-overlapping template	0.5341	1.0000
09	Overlapping template	0.7887	0.9917
10	universal	0.9496	1.0000
11	Approximate entropy	0.2993	0.9917
12	Random excursions	0.7681	0.9871
13	Random excursions Variant	0.7955	1.0000
14	Serial(P-value1)	0.8343	0.9833
	Serial(P-value2)	0.4220	0.9833
15	Linear comlexity	0.4846	1.0000
	success no.	15/15	15/15

including IWT time, anti-IWT time, encryption time and decryption time. (The IWT time refers to the time required for the integer wavelet transform of the image. The encryption time refers to the time required for the image encryption operation after n rounds of IWT. The decryption time refers to the time required for the decryption operation of the encrypted image after n rounds of IWT) From the figure, it can be seen that for small size images, the computation time of single IWT is faster, and the encryption and decryption process is relatively fast, for medium size images, the use of multiple rounds of IWT will increase the computation time, but it is still acceptable. The encryption and decryption time is relatively short and can still be completed within a reasonable time, for large size images, multiple rounds of IWT will significantly increase the computation time and the encryption and decryption time will increase accordingly, but in the case of processing large images, it can still be completed within a reasonable time frame. Therefore for large size images, by using multiple rounds of IWT can provide high real-time performance, for different size images we may need to trade-off between the IWT time and the encryption and decryption time to choose the appropriate number of IWT rounds and encryption methods. Overall, this algorithm has some advantages as an effective encryption and image processing scheme in different application scenarios.

4.10. Computational complexity analysis

In this paper, IWT is used to promote excellent image restoration and reduce computational complexity. When the size of the original image is $H \times W$, the computational complexity is $\boldsymbol{O}(HW)$, and the computational complexity can be reduced to $\boldsymbol{O}(1/4HW)$ by IWT. As shown in the following table 13, the encryption time of the algorithm in this paper is compared with the time required for encryption after DWT and the average time required for direct encryption, indicating that the time required for encryption after frequency domain transformation is significantly reduced. This experiment is simulated under matlab software, which is not sensitive to the calculation of floating point number. The time difference between IWT and DWT will be more obvious under other software.

5. Conclusion

In this paper, a chaotic image encryption scheme based on 3D S-box and IWT is proposed. Firstly, the dynamic key with plaintext correlation is generated by plaintext hash value, and the chaotic pseudo-random sequence is generated and obtained by using the key. Secondly, the plain image is decomposed into sub-bands of different frequencies by wavelet transform, and the image is mapped from the time domain to the frequency domain. Then, the bit permutation-three-dimensional S-box replace-ciphertext cross-diffusion is used to encrypt the low-frequency coefficients, and only the XOR-row and column permutation lightweight encryption is performed on the high-frequency coefficients. Finally, the final ciphertext is obtained by reconstruction. Each encryption module adopts a forward plaintext feedback encryption mechanism, which effectively enhances the avalanche effect of the password. The results show that the scheme can ensure the high-quality restoration of the

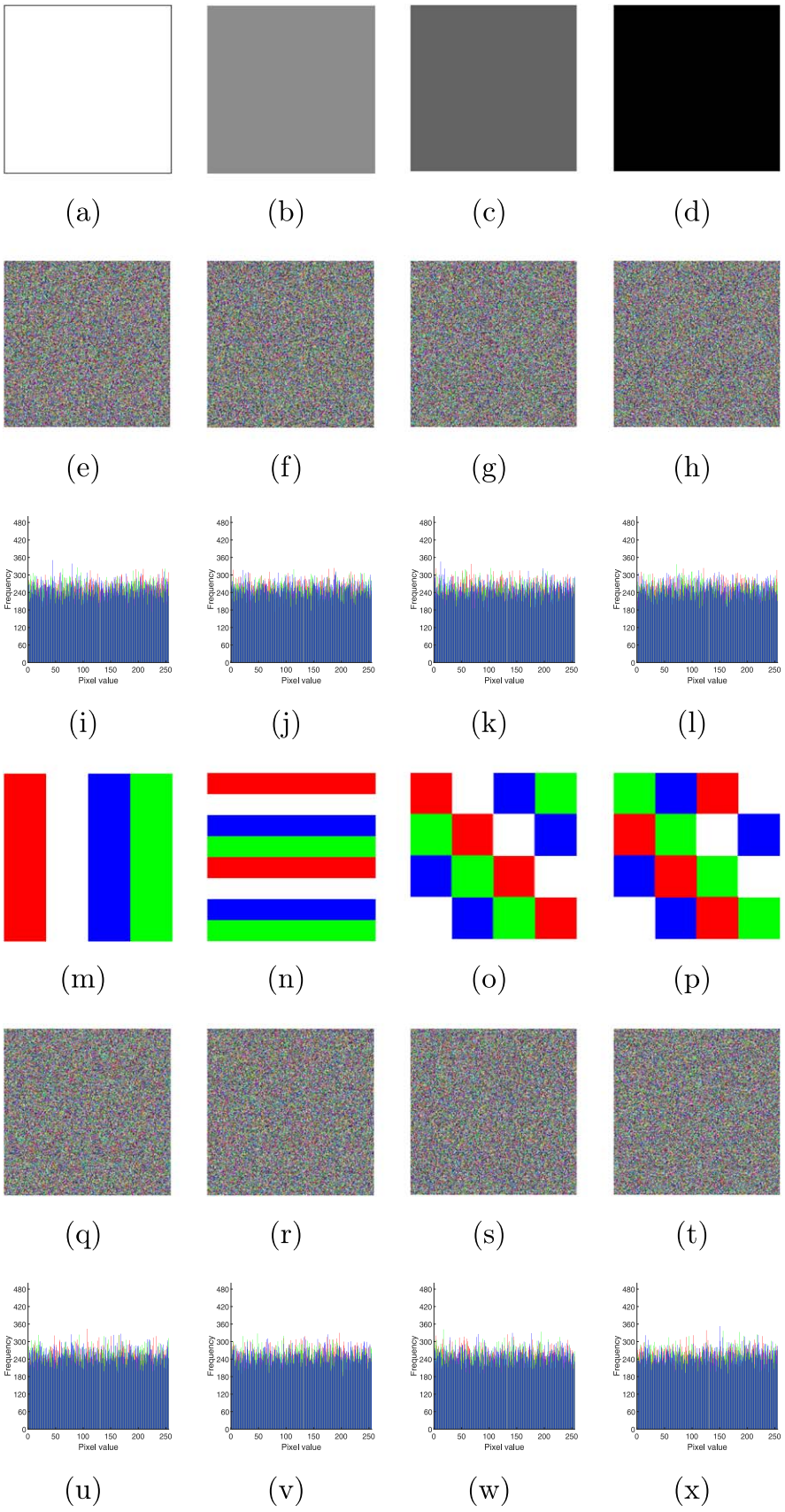

Figure 14. Specific chosen plaintext images and their corresponding attack outcomes: (a)–(d) select special total-colored images; (e)–(h) the ciphertext images corresponding to (a)–(d); (i)–(j) histogram corresponding to (e)–(h); (m)–(p) select special color images; (q)–(t) the ciphertext images corresponding to (m)–(p); (u)–(x) histogram corresponding to(q)–(t).

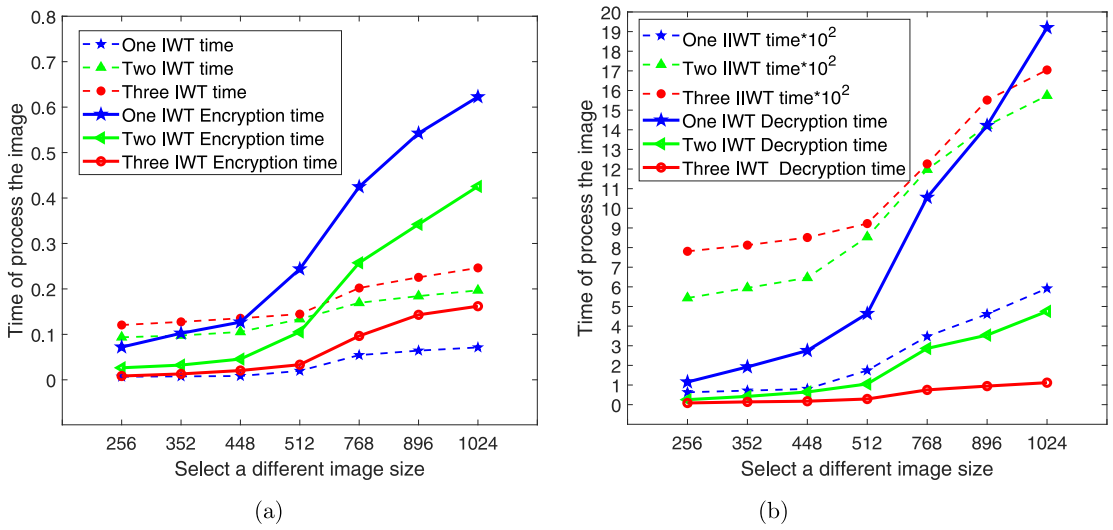

Figure 15. Time comparison of encrypting images of different sizes by N-round integer wavelet transform: (a) IWT times for multiple rounds and corresponding encryption times; (b) IIWT times for multiple rounds and corresponding decryption times.

Table 13. IWT, DWT and spatial domain of encryption time comparison.

Dimensione	256 × 256	512 × 512	1024 × 1024
IWT	0.076425s	0.302351s	1.061198s
DWT	0.079066s	0.335736s	1.089819s
Spatial domain	0.262431s	0.991714s	4.03523s

image, reduce the computational complexity, improve the real-time performance and efficiency, save the storage space, and has robustness and significant diffusion characteristics, and can successfully resist various standard password attacks. Therefore, the method proposed in this paper is considered to be effective in improving the accuracy and reliability of information exchange, especially in the context of the era of big data, which is of great significance to image encryption.

Acknowledgments

The authors acknowledge anonymous reviewers for their valuable feedback.

Data availability statement

No new data were created or analysed in this study.

Funding statement

This work was supported in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011717, and in part by Project for Zhongshan Science and Technology under Grant 2021B2062, and in part by Special Projects for Key Fields of the Education Department of Guangdong Province under Grant 2023ZDZX1041.

Conflict of interest

The authors declare, they have no conflict of interest.

CRediT author statement

Conceptualization, Z.F.; Methodology, Z.F.; Software, Z.F.; Validation, Z.F. and C.B.; Formal Analysis, H.W.; Investigation, Z.F. and Y.L.; Resources, H.W. and Y.L.; Data Curation, H.W.; Writing—Original Draft, Z.F., C. B., Y.L., H.W. and X.Z.; Writing—Review & Editing, Z.F., Y.L., C.B. and X.Z.; Visualization, Z.F. and C.B.; Supervision, H.W. and W.F.; Project Administration, H.W. and Y.L.; Funding Acquisition, H.W. All authors have read and agreed to the published version of the manuscript.

ORCID iDs

Heping Wen https://orcid.org/0000-0002-1178-4598
Zhaoyang Feng https://orcid.org/0009-0002-2238-5311
Chixin Bai https://orcid.org/0009-0004-9942-810X
Yiting Lin https://orcid.org/0000-0003-4159-3132
Wei Feng https://orcid.org/0000-0003-3023-5225

References

[1] Zhou S, Wang X and Zhang Y 2023 Novel image encryption scheme based on chaotic signals with finite-precision error *Inf. Sci.* **621** 782–98
[2] Feng W, Zhao X, Zhang J, Qin Z, Zhang J and He Y 2022 Image encryption algorithm based on plane-level image filtering and discrete logarithmic transform *Mathematics* **10** 2751
[3] Wen H, Lin Y, Kang S, Zhang X and Zou K 2024 Secure image encryption algorithm using chaos-based block permutation and weighted bit planes chain diffusion *iScience* **27** 108610
[4] Chai X, Wang Y, Chen X, Gan Z and Zhang Y 2022 Tpe-gan: Thumbnail preserving encryption based on gan with key *IEEE Signal Process Lett.* **29** 972–6
[5] Wen H, Xie Z, Wu Z, Lin Y and Feng W 2024 Exploring the future application of uavs: Face image privacy protection scheme based on chaos and dna cryptography *Journal of King Saud University—Computer and Information Sciences* **36** 101871
[6] Kocak O, Erkan U, Toktas A and Gao S 2024 Pso-based image encryption scheme using modular integrated logistic exponential map *Expert Syst. Appl.* **237** 121452
[7] Chen L, Li C and Li C 2022 Security measurement of a medical communication scheme based on chaos and dna coding *J. Visual Commun. Image Represent.* **83** 103424
[8] Zeng W, Zhang C, Liang X, Luo Y, Wang X and Qiu K 2024 Chaotic phase noise-like encryption based on geometric shaping for coherent data center interconnections *Opt. Express* **32** 1595
[9] Toktas A, Erkan U, Gao S and Pak C 2024 A robust bit-level image encryption based on bessel map *Appl. Math. Comput.* **462** 128340
[10] Wen H and Lin Y 2024 Cryptanalysis of an image encryption algorithm using quantum chaotic map and dna coding *Expert Syst. Appl.* **237** 121514
[11] Wen H, Lin Y, Xie Z and Liu T 2023 Chaos-based block permutation and dynamic sequence multiplexing for video encryption *Sci. Rep.* **13** 14721
[12] Liang X, Zhang C, Luo Y, Wang X and Qiu K 2023 Secure encryption and key management for ofdm-pon based on chaotic hilbert motion *J. Lightwave Technol.* **41** 1619–25
[13] Luo Y, Tang S, Liu J, Cao L and Qiu S 2020 Image encryption scheme by combining the hyper-chaotic system with quantum coding *Opt. Lasers Eng.* **124** 105836
[14] Li C and Yang X 2022 An image encryption algorithm based on discrete fractional wavelet transform and quantum chaos *Optik* **260** 169042
[15] Kumar Singh R, Kumar B, Kumar Shaw D and Ali Khan D 2021 Level by level image compression-encryption algorithm based on quantum chaos map *Journal of King Saud University—Computer and Information Sciences* **33** 844–51
[16] Wei D, Jiang M and Deng Y 2023 A secure image encryption algorithm based on hyper-chaotic and bit-level permutation *Expert Syst. Appl.* **213** 119074
[17] Shahna K U and Mohamed A 2020 A novel image encryption scheme using both pixel level and bit level permutation with chaotic map *Appl. Soft Comput.* **90** 106162
[18] Wang M, Wang X, Wang C, Xia Z, Zhao H, Gao S, Zhou S and Yao N 2020 Spatiotemporal chaos in cross coupled map lattice with dynamic coupling coefficient and its application in bit-level color image encryption *Chaos, Solitons Fractals* **139** 110028
[19] Wen H 2022 Design and embedded implementation of secure image encryption scheme using dwt and 2d-lasm *Entropy* **24**
[20] Koohpayeh Araghi T and Abd Manaf A 2019 An enhanced hybrid image watermarking scheme for security of medical and non-medical images based on dwt and 2-d svd *Future Gener. Comput. Syst.* **101** 1223–46
[21] Lee S-H 2014 Dwt based coding dna watermarking for dna copyright protection *Inf. Sci.* **273** 263–86
[22] Chai X, Wang Y, Chen X, Gan Z and Zhang Y 2022 Tpe-gan: Thumbnail preserving encryption based on gan with key *IEEE Signal Process Lett.* **29** 972–6
[23] Zhang Y, Zhou W, Zhao R, Zhang X and Cao X 2022 F-tpe: Flexible thumbnail-preserving encryption based on multi-pixel sum-preserving encryption *IEEE Trans. Multimedia* 1–15
[24] Zhang Y, Zhao R, Xiao X, Lan R, Liu Z and Zhang X 2022 Hf-tpe: High-fidelity thumbnail- preserving encryption *IEEE Trans. Circuits Syst. Video Technol.* **32** 947–61
[25] Wang X and Li Y 2021 Chaotic image encryption algorithm based on hybrid multi-objective particle swarm optimization and dna sequence *Opt. Lasers Eng.* **137** 106393
[26] Wen H *et al* 2022 Secure dna-coding image optical communication using non-degenerate hyperchaos and dynamic secret-key *Mathematics* **10**

[27] Wen H, Kang S, Wu Z, Lin Y and Huang Y 2023 Dynamic rna coding color image cipher based on chain feedback structure *Mathematics* **11** 3133
[28] Wang X, Liu C and Jiang D 2021 A novel triple-image encryption and hiding algorithm based on chaos, compressive sensing and 3d dct *Inf. Sci.* **574** 505–27
[29] Ariatmanto D and Ernawan F 2022 Adaptive scaling factors based on the impact of selected dct coefficients for image watermarking *Journal of King Saud University—Computer and Information Sciences* **34** 605–14
[30] Sisaudia V and Vishwakarma V P 2022 A secure gray-scale image watermarking technique in fractional dct domain using zig-zag scrambling *Journal of Information Security and Applications* **69** 103296
[31] Wen H *et al* 2023 Secure optical image communication using double random transformation and memristive chaos *IEEE Photonics J.* **15** 1–11
[32] Xie H, Lu J, Han J, Zhang Y, Xiong F and Zhao Z 2023 Fourier coded aperture transform hyperspectral imaging system *Opt. Lasers Eng.* **163** 107443
[33] Melman A and Evsutin O 2023 Comparative study of metaheuristic optimization algorithms for image steganography based on discrete fourier transform domain *Appl. Soft Comput.* **132** 109847
[34] Wen H and Lin Y 2023 Cryptanalyzing an image cipher using multiple chaos and dna operations *Journal of King Saud University—Computer and Information Sciences* **35** 101612
[35] Banerjee M, Ghosh S, Manfredi P and d'Onofrio A 2023 Spatio-temporal chaos and clustering induced by nonlocal information and vaccine hesitancy in the sir epidemic model *Chaos, Solitons Fractals* **170** 113339
[36] Liu X, Tong X, Wang Z and Zhang M 2022 A new n-dimensional conservative chaos based on generalized hamiltonian system and its' applications in image encryption *Chaos Solitons Fractals* **154** 111693
[37] Liu X, Tong X, Zhang M and Wang Z 2023 A highly secure image encryption algorithm based on conservative hyperchaotic system and dynamic biogenetic gene algorithms *Chaos Solitons Fractals* **171** 113450
[38] Liu X, Tong X, Wang Z and Zhang M 2022 A novel hyperchaotic encryption algorithm for color image utilizing dna dynamic encoding and self-adapting permutation *Multimedia Tools Appl.* **81** 21779–810
[39] Wen H, Lin Y and Feng Z 2024 Cryptanalyzing a bit-level image encryption algorithm based on chaotic maps *Engineering Science and Technology, an International Journal* (https://doi.org/10.1016/j.jestch.2024.101634)
[40] Feng W, Wang Q, Liu H, Ren Y, Zhang J, Zhang S, Qian K and Wen H 2023 Exploiting newly designed fractional-order 3d lorenz chaotic system and 2d discrete polynomial hyper-chaotic map for high-performance multi-image encryption *Fractal and Fractional* **7** 887
[41] Wen H *et al* 2023 Security analysis of a color image encryption based on bit-level and chaotic map *Multimedia Tools Appl.*
[42] Wen H, Huang Y and Lin Y 2023 High-quality color image compression-encryption using chaos and block permutation *Journal of King Saud University—Computer and Information Sciences* **page 101 660**
[43] Luo Y, Liang X, Zhang C, Zeng W and Qiu K 2024 Redundancy-free key distribution using multiple phase offset for secure data center *J. Lightwave Technol.* **42** 523–31
[44] Erkan U, Toktas A, Memiş S, Lai Q and Hu G 2023 An image encryption method based on multi-space confusion using hyperchaotic 2d vincent map derived from optimization benchmark function *Nonlinear Dynamics.* (https://doi.org/10.1007/s11071-023-08859-z)
[45] Erkan U, Toktas A and Lai Q 2023 2d hyperchaotic system based on schaffer function for image encryption *Expert Syst. Appl.* **213** 119076
[46] Chai X, Fu J, Gan Z, Lu Y, Zhang Y and Han D 2023 Exploiting semi-tensor product compressed sensing and hybrid cloud for secure medical image transmission *IEEE Internet of Things Journal* **10** 7380–92
[47] Chai X, Wang Y, Gan Z, Chen X and Zhang Y 2022 Preserving privacy while revealing thumbnail for content-based encrypted image retrieval in the cloud *Inf. Sci.* **604** 115–141
[48] Chai X, Fu J, Gan Z, Lu Y and Zhang Y 2022 An image encryption scheme based on multi-objective optimization and block compressed sensing *Nonlinear Dyn.* **108** 2671–704
[49] Gupta M D and Chauhan R K 2021 Secure image encryption scheme using 4d-hyperchaotic systems based reconfigurable pseudo-random number generator and s-box *Integration* **81** 137–59
[50] Huang L, Li W, Xiong X, Yu R, Wang Q and Cai S 2022 Designing a double-way spread permutation framework utilizing chaos and s-box for symmetric image encryption *Opt. Commun.* **517** 128365
[51] Zhu S, Deng X, Zhang W and Zhu C 2023 Secure image encryption scheme based on a new robust chaotic map and strong s-box *Math. Comput. Simul* **207** 322–46
[52] Lai Q and Liu Y 2023 A cross-channel color image encryption algorithm using two-dimensional hyperchaotic map *Expert Syst. Appl.* **223** 119923
[53] Detombe J and Tavares S 1993 *Constructing large cryptographically strong s-boxes* (Berlin: Springer) 165–81
[54] Adams C and Tavares S 1990 The structured design of cryptographically good s-boxes *J. Cryptol.* **3** 27–41
[55] Castro J C H, Sierra J M, Seznec A, Izquierdo A and Ribagorda A 2005 The strict avalanche criterion randomness test *Math. Comput. Simul* **68** 1–7
[56] Adams C and Tavares S 1990 *Good s-boxes are easy to find* (New York: Springer) 612–5
[57] Biham E and Shamir A 1991 Differential cryptanalysis of des-like cryptosystems *J. Cryptol.* 1432–378
[58] Matsui M 1994 *Linear cryptanalysis method for des cipher* (Berlin: Springer) 386–97
[59] Wang X-Y, Sun H-H and Gao H 2021 An image encryption algorithm based on improved baker transformation and chaotic s-box* *Chin. Phys. B* **30** 060507
[60] Idrees B and Zafar S 2020 Image encryption algorithm using s-box and dynamic hénon bit level permutation *Multimedia Tools Appl.* **79** 6135–62
[61] Zhu C, Hu Y and Sun K 2012 A new algorithm for image encryption based on hyper chaotic system and ciphertext interleaved diffusion *Journal of Electronics and Information* **34** 1735
[62] Shafique A and Ahmed F 2020 Image encryption using dynamic s-box substitution in the wavelet domain *Wirel. Pers. Commun.* **115** 2243–68
[63] Mansouri A and Wang X 2021 A novel block-based image encryption scheme using a new sine powered chaotic map generator *Multimedia Tools Appl.* **80** 21955–78
[64] Murillo-Escobar M A, Cruz-Hernández C, Abundiz-Pérez F, López-Gutiérrez R M and Acosta Del Campo O R 2015 A rgb image encryption algorithm based on total plain image characteristics and chaos *Signal Process.* **109** 119–31