Double-Image Visually Meaningful Encryption Algorithm Based on Compressed Sensing and Frft Embedding

donghua jiang (jiangdonghua@chd.edu.cn)
Chang'an University https://orcid.org/0000-0002-3545-6409
Lidong Liu
Chang'an University https://orcid.org/0000-0001-8142-2261
Liya Zhu
Chang'an University
Xingyuan Wang
Dalian Maritime University
Yingpin Chen
Minnan Normal University
Xianwei Rong
Harbin Normal University

Research Article

Keywords: Double-image compression and encryption, Visually meaningful encrypted image, Quantum cellular neural network, Compressed sensing, Fractional Fourier transform

Posted Date: March 17th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-291468/v1

Double-image visually meaningful encryption algorithm based on compressed sensing and FRFT embedding

Donghua Jiang[1], Lidong Liu[1]*, Liya Zhu[2], Xingyuan Wang[3], Yingpin Chen[4] and Xianwei Rong[5]

[1]*School of Information Engineering, Chang'an University, Xi'an 710064, China*

[2]*School of Electrical and Control Engineering, Chang'an University, Xi'an 710064,China*

[3]*School of Information Science and Technology, Dalian Maritime University, Da'lian 116026, China*

[4]*School of Physics and Information Engineering, Minnan Normal University, Zhang'zhou 363000, China*

[5]*Physics and Electronic Engineering School, Harbin Normal University, Harbin, 150025 China*

**Corresponding author: Lidong Liu (liulidong@chd.edu.cn)*

Abstract: The transmission of images via the Internet has grown exponentially in the past few decades. However, the Internet considered as an insecure method of information transmission may cause serious privacy issues. In order to overcome such potential security issues, a novel double-image visually meaningful encryption (DIVME) algorithm conjugating quantum cellular neural network (QCNN), compressed sensing (CS) and fractional Fourier transform (FRFT) is proposed in this paper. First, the wavelet coefficients of the two plain images are scrambled by the Fisher-Yates confusion algorithm, and then compressed by the key-controlled partial Hadamard matrix. The final meaningful cipher image is generated by embedding the encrypted images into a host image with the same resolution of the plain image via the FRFT-based embedding method. Besides, the eigenvalues of the plain images are utilized to generate the key stream to improve the ability of proposed DIVME algorithm to withstand the plaintext attacks. Afterwards, the plaintext eigenvalues are embedded into the alpha channel of the meaningful cipher image under control of the keys to reduce unnecessary storage space and transmission costs. Ultimately, the simulation results and security analyses indicate that the proposed DIVME algorithm is effective and can withstand multiple attacks.

Keywords: Double-image compression and encryption; Visually meaningful encrypted image; Quantum cellular neural network; Compressed sensing; Fractional Fourier transform

1 Introduction

With the advent of big data era, a large number of digital images are manufactured and transmitted on the Internet every moment, which will also be accompanied with various security issues [1-2]. Therefore, how to efficiently and securely transmit digital images without being intercepted and tampered is of great significance. As an effective protection method, the image encryption (IE) technology can encrypt the natural image into a noise-like image, from which any useful information cannot be visually attained by the attackers, thus playing a role in protecting image information. Additionally, the compressed sensing (CS) technology, proposed by Candes and Donoho in 2006 [3-4], is a technology that can simultaneously under-sample, compress and encrypt the sparse signals. Hence, it is not an unattractive option to design an efficient image cryptosystem by combining IE technology and CS technology.

The CS technology can be seen as a variant of symmetric encryption, in which the sparse signal, measurement matrix and sensor matrix respectively correspond to the plaintext, secret key and ciphertext in the cryptosystem from the perspective of encryption. Moreover, Rachlin *et al*. [5] pointed out that the CS-based cryptosystem has sufficient computational security to withstand the only-ciphertext attacks and brute force attacks. However, since the sensor matrix is obtained by performing linear projection on the sparse signal, it is vulnerable to the known- and chosen-plaintext attacks [6-7]. Thus,

to attain a higher security level, researchers have proposed varied image protection algorithms by combining compressed sensing with other encryption techniques, such as chaos theory [8-10], coding technology [11-12], cellular automata [13-14], neural network [15-16] and so on.

For example, a parallel image compression-encryption algorithm is presented by Huang [17]. In his scheme, first, the plain image is divided into several sub-images and then linearly measured by 1D CS. Afterwards, a series of operations such as permutation, substitution, block-wise XOR are performed on the quantized measurement value matrixes to generate the final cipher image. However, the large-scale Gaussian random measurement matrix used as the key in Huang's scheme requires additional storage space and transmission cost. Subsequently, in order to overcome this issue, key-controlled partial Hadamard matrix [18-19], partial random block weighing matrix [20], structurally random matrix [21] and chaos-based measurement matrix [22-24] are introduced to compress the plain image. Besides, in the encryption phase, the counter mode [25-26], hash function [23, 27] and plaintext eigenvalue [28] are applied to withstand the plaintext attacks, since different plain images correspond to different key streams. Nevertheless, the above-mentioned CS-based image encryption algorithm can prevent image data from leakage, but it cannot provide protection in appearance.

Therefore, Bao *et al.* [29] proposed a feasible framework for simultaneous encryption and steganography, that is encryption-embedding framework. In Bao's scheme, the plain image is first encrypted by an existing encryption algorithm to obtain a noise-like or texture-like cipher image. After that, the meaningless cipher image is decomposed and embedded into a host image by lifting wavelet transform. In the absence of compression stage, the resolution of the meaningful cipher image is four times that of the plain image, which increases the unnecessary cost of storage and transmission. Later, many improved visually meaningful image encryption algorithms [30-34] have been proposed one after another. Such as Ref.[19], where the plain image is first encrypted and compressed through the coefficient random scrambling strategy and compressed sensing with block-wise manner. Then the robust SVD embedding method is employed to embed the meaningless cipher image into the host image with the same resolution of the plain image. Besides, the counter mode is utilized to update the encryption keys to against the chosen-plaintext attacks.

In this paper, we put forward an efficient double-image visually meaningful encryption algorithm based on compressed sensing and FRFT embedding. It mainly consists of two stages: pre-encryption and embedding process. To prevent image information from leakage, in the first stage, the Fisher-Yates confusion and compressed sensing are utilized to encrypt and compress the two plain images to attain the meaningless cipher images. Afterwards, in the second stage, the meaningless cipher images are embedded into a host image via the FRFT embedding method, so that their appearance is protected. Besides, the quantum cellular neural network and improved Henon map, whose initial values are generated from the plaintext eigenvalues, are applied to construct the key-controlled measurement matrix and key streams in encryption.

The innovation and contribution of this paper are summarized as follows.

(1) An efficient double-image visually meaningful encryption algorithm based on compressed sensing and FRFT embedding is designed to improve transmission efficiency.

(2) A key-controlled double-embedding method (FRFT embedding) is proposed to improve the security of embedding phase.

(3) A novel "One cipher image corresponds to one key" mechanism is proposed to withstand the plaintext attacks.

(4) Simulation analysis and comparison results indicate that the proposed encryption scheme has high efficiency and can withstand multiple attacks.

The rest of this paper is arranged as follows. The basic knowledge related to the proposed algorithm is described in the Section 2. The third section and fourth section respectively introduce the specific steps of proposed DIVME algorithm and corresponding decryption algorithm in detail. Moreover, the simulation results and performance analysis are given in the Section 5. After that, our encryption scheme is compared with the existing related algorithms, and the results are listed in the sixth section. Then, a brief summary and future work are shown in the final section.

2 Relevant knowledge

2.1 Chaotic system

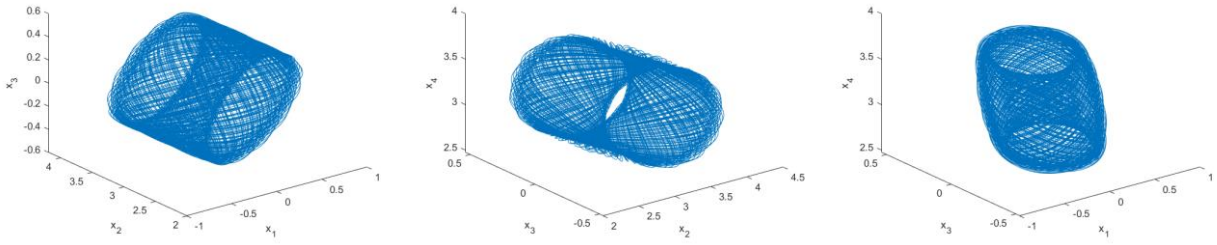
In this subsection, two chaotic systems are introduced which are hyperchaotic quantum cellular neural network and improved 2D Henon map. Among them, the quantum cellular neural network is used to construct the key streams with high unpredictability in encryption. Additionally, in order to save storage space and reduce transmission costs, a key-controlled measurement matrix is generated according to the improved Henon chaotic map.

2.1.1 Hyperchaotic quantum cellular neural network

Quantum cellular neural network (QCNN) is constructed by several quantum cellular automata (QCA) [35]. And it has complex dynamic characteristics due to quantum interaction between the quantum dots. For the two-cell QCNN, its state equation is defined as

$$\begin{cases} \dot{x}_1 = -2w_1\sqrt{1-x_1^2}\sin x_2 \\ \dot{x}_2 = -w_2(x_1 - x_3) + 2w_1\dfrac{x_1}{\sqrt{1-x_1^2}}\cos x_2 \\ \dot{x}_3 = -2w_3\sqrt{1-x_3^2}\sin x_4 \\ \dot{x}_4 = -w_4(x_3 - x_1) + 2w_3\dfrac{x_3}{\sqrt{1-x_3^2}}\cos x_4 \end{cases} \tag{1}$$

Where x_1 and x_3 are polarizations. x_2 and x_4 are the quantum phase. Moreover, w_1 and w_3 are the proportional coefficients of inter-dot energy in each cell. w_2 and w_4 are the weighted impact coefficients of the difference in the polarizations of adjacent cells. Additionally, when $w_1 = w_3 = 0.28$, $w_2 = 0.7$ and $w_4 \in (0.1, 1]$, Eq.(1) enters the hyper-chaotic state, and its corresponding hyperchaotic trajectories are plotted in Fig.1.



(a) x_1-x_2-x_3 (b) x_2-x_3-x_4 (c) x_1-x_3-x_4

Fig.1 The hyperchaotic trajectories of this quantum cellular neural network.

2.1.2 Improved 2D Henon map

Since the classical 2D Henon map has a small key space and its chaotic trajectories are simple, an improved Henon map (IHM) is proposed in Ref.[36]. Its system equation is as follows.

$$\begin{cases} u_{n+1} = (1 - au_n^2)\sin v_n \\ v_{n+1} = \sin bv_n + u_n \end{cases} \tag{2}$$

Where a and b are system control parameters. Furthermore, u_{n+1} and v_{n+1} are the generated pseudo-random numbers, belonging to [-1, 1]. Fig.2 displays the bifurcation diagram of the variable u under the condition that a is set to 2.1 and the

initial value is $[0.53, 0.89]^T$. As Fig.2 shows, when $b \in (-\infty, -0.74] \cup [0.74, +\infty)$, the improved Henon map is in a chaotic state.

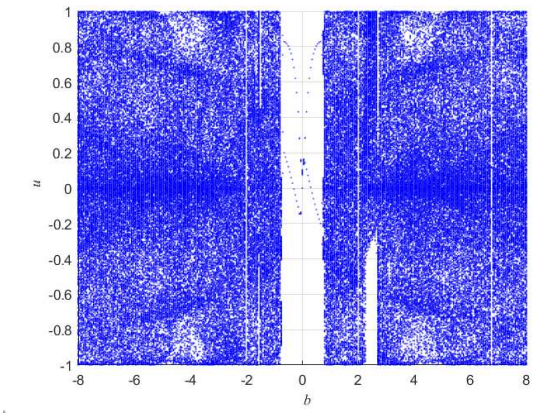


Fig.2 The bifurcation diagram of improved Henon map.

2.2 Compressed sensing

Compressed sensing [3-4] refers to using a measurement matrix unrelated to the transformation basis to linearly project the sparse high-dimensional signals into a low-dimensional space, and then reconstructing the original signals with high probability from these few projections.

Suppose that the natural signal $\boldsymbol{y} = \{y_1, y_2, y_3, ..., y_N\}$ can be expressed linearly by a group of orthonormal basis, as shown in Eq.(3).

$$\boldsymbol{y} = \boldsymbol{\Psi S} = \sum_{i=1}^{N} S_i \boldsymbol{\Psi}_i \tag{3}$$

In Eq.(3), $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \boldsymbol{\Psi}_3, ..., \boldsymbol{\Psi}_N]$ is the basis matrix. And the column vector $\boldsymbol{S}$ sized of $N \times 1$ is the sparse representation coefficient of $\boldsymbol{y}$ in $\boldsymbol{\Psi}$. Besides, if $\|\boldsymbol{S}\|_0 = k$, $\boldsymbol{y}$ is said to be k-sparse on the orthonormal basis $\boldsymbol{\Psi}$. Then the process of linearly measuring the signal $\boldsymbol{y}$ with sparsity through the measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ can be expressed as

$$\boldsymbol{z} = \boldsymbol{\Phi y} = \boldsymbol{\Phi \Psi S} = \boldsymbol{\Theta S} \tag{4}$$

Where $\boldsymbol{z} = \{z_1, z_2, z_3, ..., z_M\}$ is the observed vector, and $\boldsymbol{\Theta} = \boldsymbol{\Phi \Psi}$ is called the sensing matrix.

Since Eq.(4) is an underdetermined equation system, other regular constraints need to be added to restore the natural signal $\boldsymbol{y}$. Related studies [37] indicate that if the signal $\boldsymbol{y}$ is sparse enough on the orthogonal basis $\boldsymbol{\Psi}$, and when the matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are irrelevant, the sparse coefficient vector $\boldsymbol{S}$ can be recovered from the vector $\boldsymbol{z}$ with high probability by solving the convex optimization problem, shown in Eq.(5). Finally, the inverse transform of sparse representation is performed on the vector $\boldsymbol{S}$ to restore the natural signal $\boldsymbol{y}$.

$$\widehat{\boldsymbol{S}} = argmin \|\boldsymbol{S}\|_0, \quad s.t. \quad \boldsymbol{z} = \boldsymbol{\Theta S} \tag{5}$$

In Eq.(5), $\|\boldsymbol{S}\|_0$ refers to the l_0-norm of vector $\boldsymbol{S}$, which is equal to the number of non-zero elements in the vector.

2.3 Fractional Fourier transform

The fractional Fourier transform (FRFT) is a generalized form of the Fourier transform [38-39]. It is obtained by rotating the natural signal counterclockwise at any angle on the time axis. Therefore, the FRFT of a signal contains both its time- and frequency-domain features. The $p1$, $p2$-order FRFT of a two-dimensional signal $f(x, y)$ is defined as follows.

$$F^{p1,\,p2}(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) k_{p1,\,p2}(x, y, u, v) dxdy \tag{6}$$

In Eq.(6), the kernel function $k_{p1,\,p2}(x, y, u, v) = \frac{\sqrt{1-i\cot\alpha}\sqrt{1-i\cot\beta}}{2\pi} \exp\left[\frac{i(x^2+u^2)}{2\tan\alpha} - \frac{ixu}{\sin\alpha}\right] \exp\left[\frac{i(y^2+v^2)}{2\tan\beta} - \frac{iyv}{\sin\beta}\right]$, where α and β represent the rotation angles on the x and y axes, respectively.

2.4 Fisher-Yates confusion

Fisher-Yates algorithm, also known as Knuth random scrambling algorithm, is utilized to scramble the sparse coefficient matrix of plain image to reduce the strong correlation between adjacent sparse coefficients. Its scrambling process is illustrated in Fig.3. In operation, the chaotic sequence generated by the QCNN is used to replace each randomly generated number, effectively controlling the elements exchanged each time. Meanwhile, the initial value of the QCNN is calculated by extracting partial pixels of the plain image with the secret keys. Therefore, different plain images correspond to different scrambling sequences.

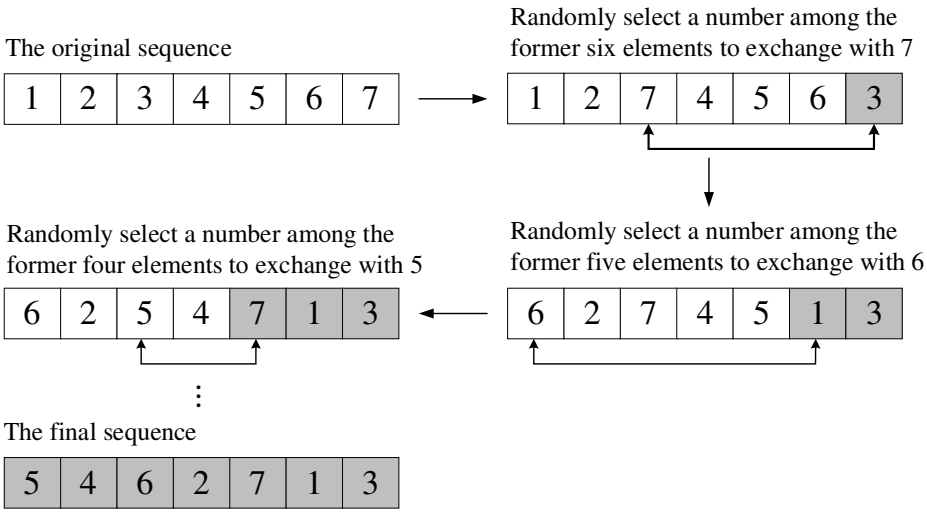


Fig.3 The scrambling process of Fisher-Yates algorithm.

3 The proposed encryption scheme

3.1 Generation of some important parameters

3.1.1 Computing the initial values for QCNN

Logistic map is utilized to extract partial pixels of plain images ***P1*** and ***P2*** to generate the initial values of the QCCN, assuming that their resolutions are both $m \times n$. Its definition is displayed in Eq.(7). When the system parameter μ is between 3.57 and 4, this map exhibits chaotic characteristics.

$$r_n = \mu \times r_{n-1}(1 - r_{n-1}),\ n = 2, 3, ... \tag{7}$$

The detailed approach for generating the initial values of the QCNN is displayed as follows.

Step 1. First, the Logistic map is iterated ($mn+T_0$) times with the initial value r_0, and the first T_0 elements are discarded to obtain a chaotic sequence $\boldsymbol{r} = \{r_1, r_2, r_3, ..., r_{mn}\}$. Then the sequence $\boldsymbol{r}$ is sorted in ascending order to generate a new sequence $\boldsymbol{Tr}$.

Step 2. Calculate the sum of $\left[\boldsymbol{P1}_{\boldsymbol{Tr}(1)}, \boldsymbol{P1}_{\boldsymbol{Tr}(2)}, ..., \boldsymbol{P1}_{\boldsymbol{Tr}(mn/2)}\right]$ and $\left[\boldsymbol{P2}_{\boldsymbol{Tr}(mn/2+1)}, \boldsymbol{P2}_{\boldsymbol{Tr}(mn/2+2)}, ..., \boldsymbol{P2}_{\boldsymbol{Tr}(mn)}\right]$ and its average value av_1 is determined. The mathematical description of this step is shown as follows.

$$su = \sum_{i=1}^{mn/2} [\boldsymbol{P1}(\boldsymbol{Tr}_i) + \boldsymbol{P2}(\boldsymbol{Tr}_{mn/2+i})] \tag{8}$$

$$av_1 = \frac{su}{m \times n} \tag{9}$$

Step 3. Finally, the initial values of the QCNN are calculated by performing Eq.(11). Where *mod* means modular operation, $\lfloor x \rfloor$ represents an integer not greater than x, and k_i $(i = 1, 2)$ is the external key parameter.

$$fv = \frac{\lfloor (av_1 - \lfloor av_1 \rfloor) \times 255 \rfloor}{255} \tag{10}$$

$$\begin{cases} \dot{x}_1 = fv + k_1 \bmod 1 \\ \dot{x}_2 = fv + \dot{x}_1 \bmod 2 \ + \ 2 \\ \dot{x}_3 = fv \times 10^3 + k_2 \bmod 1 \\ \dot{x}_4 = fv \times 10^3 + \dot{x}_3 \bmod 2 \ + \ 2 \end{cases} \tag{11}$$

3.1.2 Obtaining the initial values for IHM

Storing the entire measurement matrix directly requires a lot of space, and sufficient bandwidth needs to be used to transmit it to the decoder. Thus, in this paper, the measurement matrix is determined by the chaotic sequence generated by the improved Henon map. At the same time, to improve the anti-attack ability of the algorithm, partial content of two plain images will be used to generate the initial values of the IHM.

The process of generating the initial values of the improved Henon map is as follows.

Step 1. First, the average value av_2 is determined as follows.

$$av_2 = \frac{1}{m \times n} \sum_{i=1}^{mn/2} [\boldsymbol{P1}(\boldsymbol{Tr}_{mn/2+i}) + \boldsymbol{P2}(\boldsymbol{Tr}_i)] \tag{12}$$

Step 2. Then the parameter gv is obtained by performing Eq.(13) on the average value av_2. Where len(x) represents the number of integer parts of x. For example, $\text{len}(153.72902) = 3$.

$$gv = \frac{\lfloor (av_2 \times 10^{-\text{len}(av_2)}) \times 255 \rfloor}{255} \tag{13}$$

Step 3. Finally, the initial values of the IHM are computed according to the following equation. Where k_3 and k_4 are the external key parameters. Additionally, sign(·) means sign function.

$$\begin{cases} u_0 = (\log_{k_3} gv \bmod 1) \times \text{sign}(gv - k_3) \\ v_0 = u_0 + k_4 \bmod 1 \end{cases} \tag{14}$$

3.2 The DIVME algorithm

The flow chart of the proposed DIVME algorithm is displayed in Fig. 4. As Fig.4 shows, it mainly consists of two stages. In the first stage, the secret information carried by two plain images is encrypted and compressed by the Fisher-Yates confusion and the key-controlled partial Hadamard matrix, respectively. Then, in the second stage, on the one hand, the encrypted data is randomly embedded into the host image through the fractional Fourier transform embedding, and this process is controlled by the index sequence generated from the QCNN. In addition, some important parameters are also hidden in the alpha channel of the visually meaningful cipher image.

3.2.1 Pre-encryption process

Step 1. First, an orthogonal sparse representation matrix $\boldsymbol{\Psi} \in \mathbb{R}^{n \times m}$ is constructed using the Daubechies wavelet. Then, the sparse processing is performed on two plain images ***P1*** and ***P2*** through Eq.(15), where the symbol $\boldsymbol{\Psi}^{\mathrm{T}}$ represents the transpose matrix of $\boldsymbol{\Psi}$.

$$\begin{cases} \boldsymbol{P3} = \boldsymbol{\Psi} \times \boldsymbol{P1} \times \boldsymbol{\Psi}^{\mathrm{T}} \\ \boldsymbol{P4} = \boldsymbol{\Psi} \times \boldsymbol{P2} \times \boldsymbol{\Psi}^{\mathrm{T}} \end{cases} \tag{15}$$

Step 2. To further improve the sparsity of coefficient matrices ***P3*** and ***P4***, the elements whose absolute values are less than or equal to the threshold values *Ts1* and *Ts2* are forced to be set to zero. And the matrices after threshold processing are denoted as ***P5*** and ***P6*** respectively.

Step 3. After sparse processing, most of the energy of two plain images ***P1*** and ***P2*** is mainly concentrated in the upper left corner of the matrices ***P5*** and ***P6***, which is not conducive to parallel compression. Thus, the Fisher-Yates confusion is utilized to evenly distribute the energy to the entire matrix. The QCNN is iterated ($mn+T_0$) times with the initial value $[\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4]^{\mathrm{T}}$. Then four chaotic sequences with size of $1 \times mn$ are obtained by abandoning the former T_0 values, as shown in Eq.(16).

$$\begin{cases} \boldsymbol{X} = \{x_1, x_2, x_3, ..., x_{mn}\} \\ \boldsymbol{Y} = \{y_1, y_2, y_3, ..., y_{mn}\} \\ \boldsymbol{Z} = \{z_1, z_2, z_3, ..., z_{mn}\} \\ \boldsymbol{W} = \{w_1, w_2, w_3, ..., w_{mn}\} \end{cases} \tag{16}$$

Step 4. Then, the random sequences $\boldsymbol{X}$ and $\boldsymbol{Y}$ are processed according to the following equation.

$$\begin{cases} \boldsymbol{Tx} = \lfloor \boldsymbol{X} \times 10^{10} \rfloor \bmod [mn: -1: 1] + 1 \\ \boldsymbol{Ty} = \lfloor \boldsymbol{Y} \times 10^{10} \rfloor \bmod [mn: -1: 1] + 1 \end{cases} \tag{17}$$

Step 5. As described in Section 2.4, the Fisher-Yates algorithm is used to scramble the matrices ***P5*** and ***P6***, which is controlled by the sequences $\boldsymbol{Tx}$ and $\boldsymbol{Ty}$. After confusion, the resulting matrices are named ***P7*** and ***P8*** respectively.

Step 6. Next, the measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{cn \times m}$ is constructed using the hardware-friendly Hadamard matrix, where $cn = CR \times n$. *CR* is the preset compression rate. The IHM is iterated ($mn+T_0$) times under the condition that the system parameter and the initial value are set to [2.1, 6.74] and $[u_0, v_0]^{\mathrm{T}}$, respectively. Then, the chaotic sequence $\boldsymbol{U} = \{u_{T_0+1}, u_{T_0+2}, u_{T_0+3}, ..., u_{T_0+mn}\}$ is determined.

Step 7. Construct a Hadamard matrix $\boldsymbol{H}$ sized of $m \times m$. It is calculated by the Kronecker product of two low-order matrices, and its recursion equation is shown in Eq.(18). Additionally, considering the constraints of the Hadamard matrix, we assume that *m* can be divisible by four.

$$\boldsymbol{H} = \boldsymbol{H}_2 \otimes \boldsymbol{H}_{2^{n-1}} = \begin{bmatrix} \boldsymbol{H}_{2^{n-1}} & \boldsymbol{H}_{2^{n-1}} \\ \boldsymbol{H}_{2^{n-1}} & -\boldsymbol{H}_{2^{n-1}} \end{bmatrix} \tag{18}$$

Where $\boldsymbol{H}_2 = \frac{\sqrt{2}}{2}\begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$.

Step 8. Sort the sequence $\boldsymbol{U}$ in ascending order to generate the index sequence $\boldsymbol{Tu}$. Then, the key-controlled partial Hadamard matrix $\boldsymbol{\Phi}$ is obtained by the following equation.

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{H}(\boldsymbol{Tu}_1, :) \\ \boldsymbol{H}(\boldsymbol{Tu}_2, :) \\ \boldsymbol{H}(\boldsymbol{Tu}_3, :) \\ \vdots \\ \boldsymbol{H}(\boldsymbol{Tu}_{cn}, :) \end{bmatrix} \tag{19}$$

Step 9. The matrices ***P7*** and ***P8*** are compressed in parallel by the key-controlled measurement matrix $\boldsymbol{\Phi}$ to generate the encrypted matrices ***P9*** and ***P10***. This process can be described by Eq.(20).

$$\boldsymbol{P}(i+2) = \boldsymbol{\Phi} \times \boldsymbol{P}i, \quad i = 7, 8 \tag{20}$$

3.2.2 Embedding process

Step 1. Select a host image $\boldsymbol{HI} \in \mathbb{N}^{m \times n}$ and perform the 2D discrete cosine transform on it, which is formulated in Eq.(21).

$$\boldsymbol{H1}(i, j) = \frac{2}{\sqrt{mn}} c(i) c(j) \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} \boldsymbol{HI}(p, q) \cos \frac{(2p+1) i\pi}{2m} \cos \frac{(2q+1) j\pi}{2n} \tag{21}$$

Where the transform kernel function $c(i) = c(j) = \begin{cases} \frac{1}{\sqrt{2}}, & i = 0, j = 0 \\ 1, & otherwise \end{cases}$.

Step 2. The sub-matrix ***H2*** is determined by Eq.(22). And then apply the 2D fractional Fourier transform on it with the rotation angle $[\alpha, \beta]$ to get the complex matrix $\boldsymbol{H3} = \boldsymbol{RP1} + \boldsymbol{IP1}\zeta$. ζ is the imaginary unit.

$$\boldsymbol{H2} = \begin{bmatrix} \boldsymbol{H1}(m/2+1,\ n/2+1) & \cdots & \boldsymbol{H1}(m/2+1, n) \\ \vdots & \ddots & \vdots \\ \boldsymbol{H1}(m,\ n/2+1) & \cdots & \boldsymbol{H1}(m, n) \end{bmatrix} \tag{22}$$

Step 3. By sorting the chaotic sequences $[\boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{Z}_3, ..., \boldsymbol{Z}_{cn \times n}]$ and $[\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3, ..., \boldsymbol{W}_{cn \times n}]$ in ascending order, the index sequences $\boldsymbol{Tz}$, and $\boldsymbol{Tw}$ are acquired. Then the index confusion is carried out on the ***P9*** and ***P10*** according to Eq.(23), where $i = 1, 2, ..., cn \times n$.

$$\begin{cases} \boldsymbol{P11}(i) = \boldsymbol{P9}(\boldsymbol{Tz}(i)) \\ \boldsymbol{P12}(i) = \boldsymbol{P10}(\boldsymbol{Tw}(i)) \end{cases} \tag{23}$$

Step 4. The matrices ***P11*** and ***P12*** are embedded into the real and imaginary parts of the complex matrix ***H3***, respectively, after adjusting their amplitudes by the gain factor γ. These operations are formulated by Eq.(24).

$$\begin{cases} \boldsymbol{RP2} = \boldsymbol{RP1} + (1-\gamma) \times \boldsymbol{P11} \\ \boldsymbol{IP2} = \boldsymbol{IP1} + (1-\gamma) \times \boldsymbol{P12} \end{cases} \tag{24}$$

Step 5. Then, perform the inverse 2D fractional Fourier transform on the complex matrix $\boldsymbol{H4} = \boldsymbol{RP2} + \boldsymbol{IP2}\zeta$ and replace the lower right corner of matrix ***H1*** to obtain a new complex matrix ***H5***.

Step 6. Next, the complex matrix $\boldsymbol{H6} = \boldsymbol{RP3} + \boldsymbol{IP3}\zeta$ is generated by applying the inverse 2D discrete cosine transform on the ***H5***.

Step 7. To facilitate storage, the matrix ***IP3*** is processed by the following equation. Then the ***Alp*** is used as the alpha channel together with the matrix ***RP3*** to generate the final visually meaningful cipher image $\boldsymbol{CI} \in \mathbb{N}^{m \times n}$.

$$\boldsymbol{Alp} = 247 + \boldsymbol{IP3} \tag{25}$$

Step 8. In the proposed scheme, there are different parameters (fv and gv) for different plain images. Thus, taking into account the basic principles of symmetric encryption, we embed them into the alpha channel of the cipher image. See Eq.(26) for the specific operation. At this point, the entire encryption process is complete.

$$\begin{cases} \boldsymbol{Alp}(\lfloor h_1 \times 10^{10} \rfloor \bmod [m,n]+1) = fv \times 255 \\ \boldsymbol{Alp}(\lfloor h_2 \times 10^{10} \rfloor \bmod [m,n]+1) = gv \times 255 \end{cases} \tag{26}$$

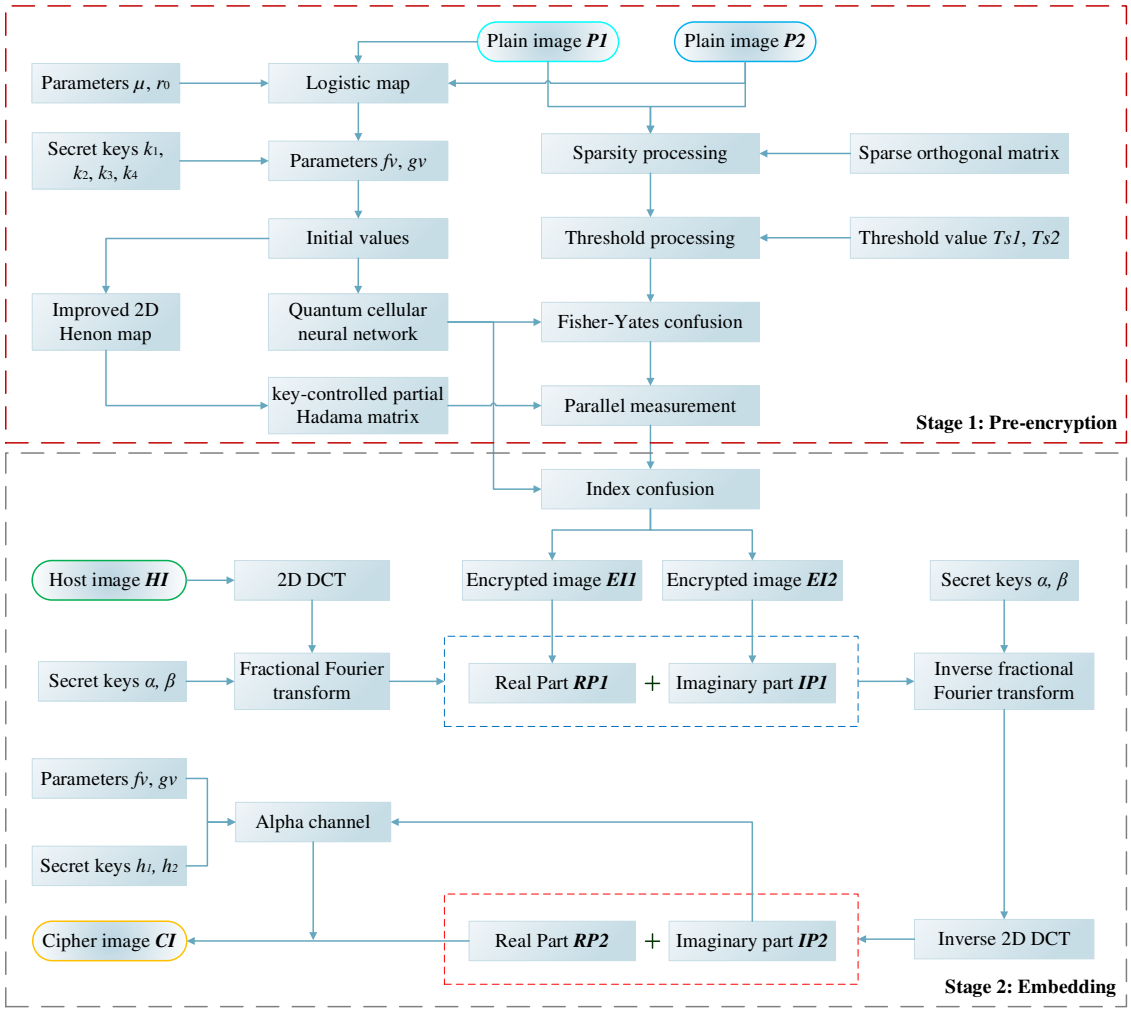


Fig.4 The schematic of the proposed DIVME algorithm.

4 The image decryption algorithm

The inverse process of DIVME algorithm is the decryption algorithm, which also includes two processes, namely the extraction and decryption process. To successfully decrypt the plain images, some external secret keys need to be transmitted to the decoder through a private channel, including k_i $(i = 1, 2, 3, 4)$, α, β, γ and h_i $(i = 1, 2)$. Meanwhile, the internal secret keys *fv* and *gv* are extracted from the alpha channel of the visually meaningful cipher image. Besides, the host image is indispensable for extracting the encrypted data from the cipher image. Thus, it is recommended to randomly select the host image from a public database to avoid increasing extra transmission costs. Then, the detailed decryption process is shown below.

4.1 Extraction process

Step 1. the parameters *fv* and *gv* are extracted from the alpha channel of the cipher image according to Eq.(27). Then, to avoid affecting the quality of the decrypted images, the values of these two parameters in the ***Alp*** are set to 247 after extraction.

$$\begin{cases} fv = \frac{\boldsymbol{Alp}(\lfloor h_1 \times 10^{10} \rfloor \bmod [m, n]+1)}{255} \\ gv = \frac{\boldsymbol{Alp}(\lfloor h_2 \times 10^{10} \rfloor \bmod [m, n]+1)}{255} \end{cases} \tag{27}$$

Step 2. By performing Eq.(11) and Eq.(14), the initial values of the QCNN and the IHM are obtained. And then iterate them to generate the key streams ***X***, ***Y***, ***Z***, ***W*** and ***U***.

Step 3. The modified matrix ***Alp*** is processed by the following equation to extract the ***IP3***.

$$\boldsymbol{IP3} = \boldsymbol{Alp} - 247 \tag{28}$$

Step 4. The complex matrix ***H5*** is determined by carrying out the 2D discrete cosine transform on the $\boldsymbol{H6} = \boldsymbol{RP3} + \boldsymbol{IP3}\zeta$.

Step 5. After performing the 2D fractional Fourier transform on the $\boldsymbol{H4}$, The matrices ***P11*** and ***P12*** are obtained according to the Eq.(29).

$$\begin{cases} \boldsymbol{P11} = \frac{\boldsymbol{RP2}-\boldsymbol{RP1}}{(1-\gamma)} \\ \boldsymbol{P12} = \frac{\boldsymbol{IP2}-\boldsymbol{IP1}}{(1-\gamma)} \end{cases} \tag{29}$$

Step 6. After inverse scrambling of the matrices ***P11*** and ***P12*** with the index sequences $\boldsymbol{Tz}$ and ***Tw*** respectively, the encrypted matrices ***P9*** and ***P10*** are generated.

4.2 Decryption process

Step 1. The key-controlled measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{cn \times m}$ is generated as described in the step 8 of section 3.2.1, And then utilized to recover the matrices ***P7*** and ***P8*** with the SL_0 algorithm. It can be denoted as Eq.(30).

$$\boldsymbol{P}(i-2) = SL_0(\boldsymbol{\Phi}, \boldsymbol{P}i), \quad i = 7, 8 \tag{30}$$

Step 2. Inverse Fisher-Yates confusion (IFYC) is applied to the matrices ***P7*** and ***P8*** with the ***Tx*** and ***Ty*** generated by sorting the sequences ***X*** and ***Y***, returning the matrices ***P5*** and ***P6***, respectively.

Step 3. By performing inverse sparse transform (see Eq.(31)) on the ***P5*** and ***P6***, then the plain images ***P1*** and ***P2*** with resolution of $m \times n$ are reconstructed.

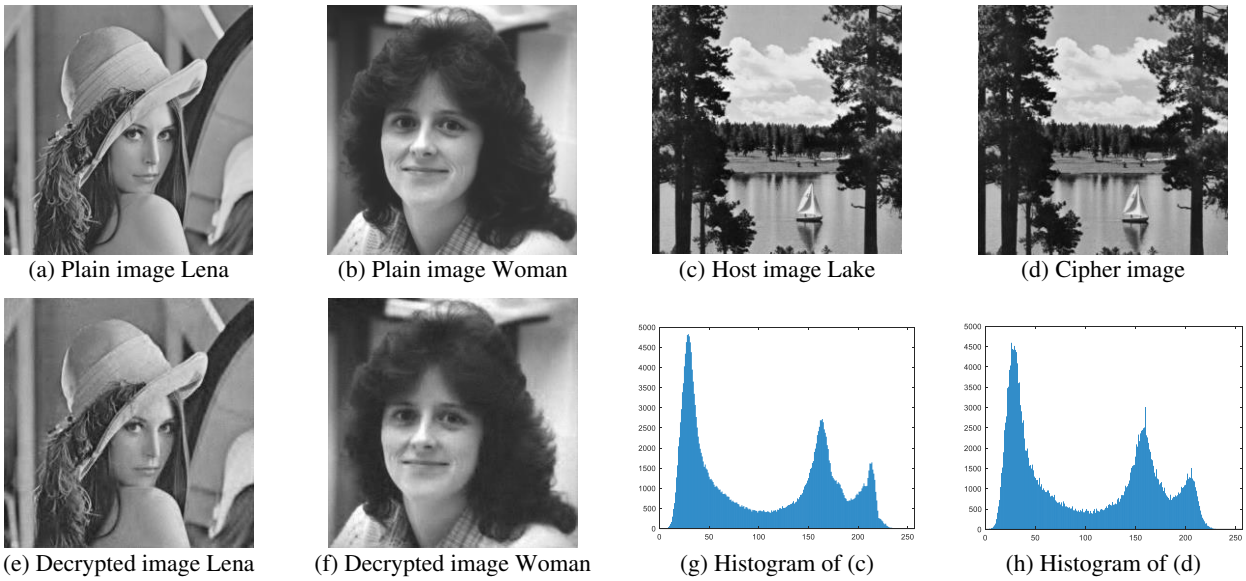
$$\begin{cases} \boldsymbol{P1} = \boldsymbol{\Psi}^{\mathrm{T}} \times \boldsymbol{P5} \times \boldsymbol{\Psi} \\ \boldsymbol{P2} = \boldsymbol{\Psi}^{\mathrm{T}} \times \boldsymbol{P6} \times \boldsymbol{\Psi} \end{cases} \tag{31}$$

5 Simulation results and performance analyses

5.1 Simulation environment setting

The simulation experiments are all carried out on the laptop computer equipped with 1.8GHz i7-8550U CPU and 16G RAM. The operating system is Microsoft Windows 10, and the simulation platform is selected as Matlab 2018b. Three

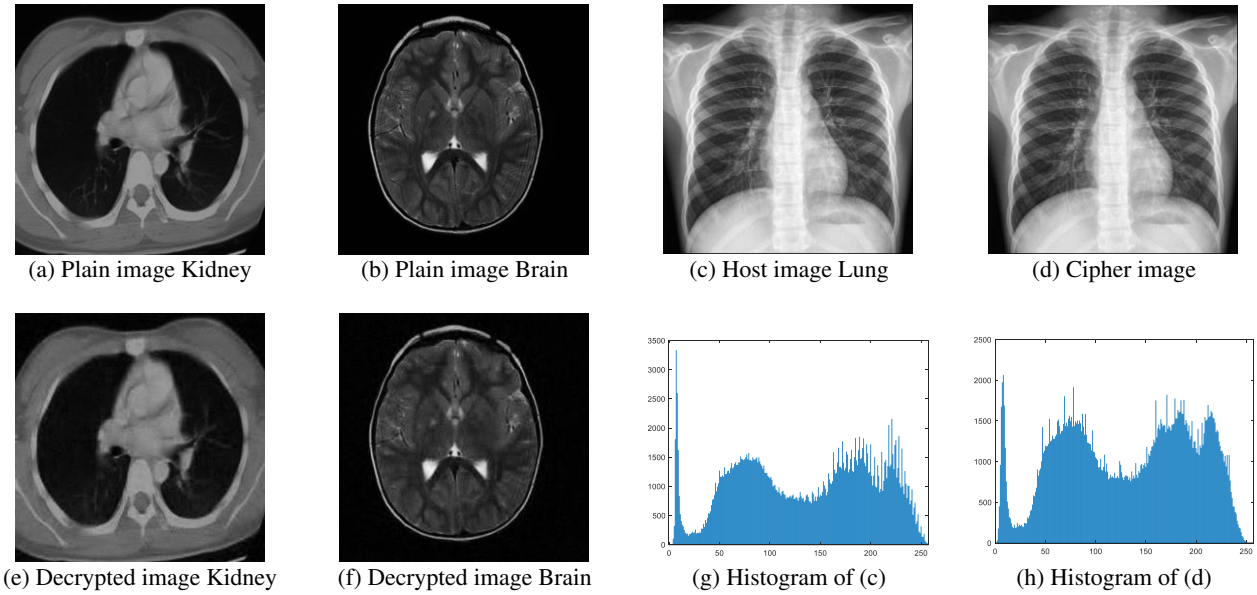
groups of plain images with different resolutions are used for the test images. Meanwhile, the external secret keys are set as follows: $k_1 = 0.9723, k_2 = 0.7875, k_3 = 0.3746, k_4 = 0.2481, h_1 = 0.2849, h_2 = 0.7128, \alpha = 1.41, \beta = 0.6$ and $\gamma = 0.998$. Additionally, the remaining encryption parameters are $Ts1 = 35, Ts2 = 25, CR = 0.25, r_0 = 0.4751, \mu = 3.99$. And in the encryption phase, the chaotic systems first iterate 500 times to eliminate the transient effect.



(a) Plain image Lena (b) Plain image Woman (c) Host image Lake (d) Cipher image

(e) Decrypted image Lena (f) Decrypted image Woman (g) Histogram of (c) (h) Histogram of (d)

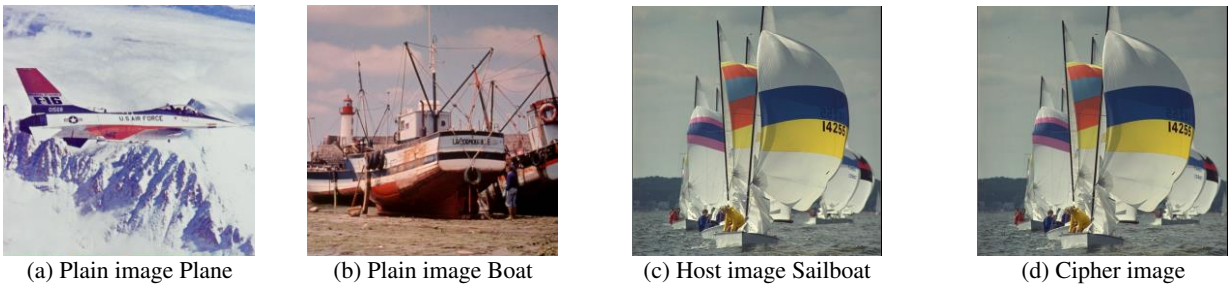
Fig.5 Simulation results with Lena and Woman as plain images (resolution 512×512).



(a) Plain image Kidney (b) Plain image Brain (c) Host image Lung (d) Cipher image

(e) Decrypted image Kidney (f) Decrypted image Brain (g) Histogram of (c) (h) Histogram of (d)

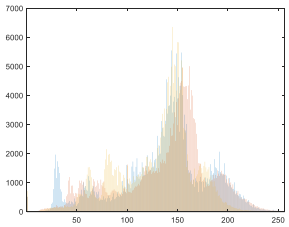
Fig.6 Simulation results with Kidney and Brain as plain images (resolution 512×512).



(a) Plain image Plane (b) Plain image Boat (c) Host image Sailboat (d) Cipher image


(e) Decrypted image Plane


(f) Decrypted image Boat


(g) Histogram of (c)

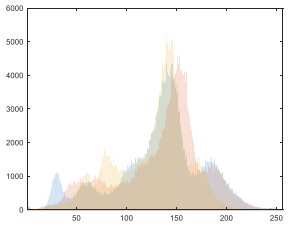

(h) Histogram of (d)

Fig.7 Simulation results with Plane and Boat as plain images (resolution $512 \times 512 \times 3$).

5.2 Encryption and decryption results

To demonstrate the effectiveness and practicability of the proposed DIVME algorithm, the simulation experiments are conducted in this subsection, and the results are drawn in the Fig.5-Fig.7. As can be seen, the plain images are encrypted into the same resolution cipher images which are meaningful and visually similar to the corresponding host images, indicating that the proposed FRFT embedding method is effective. Actually, when the visually meaningful cipher image is transmitted or stored together with other natural images, it is less likely to be discovered and attacked by the attackers, making it more secure. In other respects, the decrypted images are visually identical to their respective plain images.

Next, to quantitatively analyze the imperceptibility of cipher images and the quality of decrypted images, the peak signal-to-noise ratio (PSNR) [40] and mean structural similarity (MSSIM) [41] will be employed, which are defined in Eq.(32) and Eq.(33) respectively. Then the PSNR and MSSIM values of simulation results are listed in Tab.1. As illustrated in Tab.1, regardless of the resolution, the values of $PSNR_{cip}$ are greater than 34 dB, and $MSSIM_{cip}$ are approximately equal to one. Thus, these numerical results indicate that the encrypted data carried by the cipher images is highly invisible. Moreover, the $PSNR_{dec}$ are basically greater than 30 dB. And as the resolution of the plain image increases, the quality of the decrypted image also increases. To a certain extent, the reconstruction quality is satisfactory. In short, the proposed encryption scheme can provide double protection of image data and appearance, and has great potential for application in other fields such as medicine and transportation.

$$PSNR = 10 \times log \frac{255^2}{\frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (\boldsymbol{I}_1^{i,j} - \boldsymbol{I}_2^{i,j})^2} \quad \text{(dB)} \tag{32}$$

$$MSSIM = \frac{1}{L} \times \sum_{i=1}^{L} \frac{2\mu_{\mathbf{I}_1}^i \mu_{\mathbf{I}_2}^i + C_1}{(\mu_{\mathbf{I}_1}^i)^2 + (\mu_{\mathbf{I}_2}^i)^2 + C_1} \times \frac{2\sigma_{\mathbf{I}_1}^i \sigma_{\mathbf{I}_2}^i + C_2}{(\sigma_{\mathbf{I}_1}^i)^2 + (\sigma_{\mathbf{I}_2}^i)^2 + C_2} \times \frac{\sigma_{\mathbf{I}_1\mathbf{I}_2}^i + C_3}{\sigma_{\mathbf{I}_1}^i \times \sigma_{\mathbf{I}_2}^i} \tag{33}$$

Where $C_1 = (0.01 \times 255)^2$, $C_2 = (0.03 \times 255)^2$ and $C_3 = C_2 \times 0.5$. Besides, $\mu_{\mathbf{I}_1}^i$ and $\sigma_{\mathbf{I}_1}^i$ respectively represent the mean value and variance of the *i*-th non-overlapping block in the image $\mathbf{I}_1$.

Tab.1 PSNR and MSSIM values of simulation results.

Resolution	Plain image	Host image	$PSNR_{dec}$ (dB)	$PSNR_{cip}$ (dB)	$MSSIM_{cip}$
256×256	Finger	Baboon	30.2669	34.6253	0.9903
	Cameraman		30.1529		
	Peppers	Goldhill	29.7925	35.4728	0.9844
	Zelda		30.1743		
512×512	Lena	Lake	32.9445	34.3542	0.9970
	Woman		34.4590		
	Kidney	Lung	34.3181	34.1042	0.9983
	Brain		36.1321		

*Where $PSNR_{cip}$ and $MSSIM_{cip}$ mean the PSNR and MSSIM values between the host image and the cipher image,

respectively. Additionally, $PSNR_{dec}$ is the PSNR value among the plain image and the decrypted image.

5.3 Influence of gain factor on simulation results

Considering that the amplitude of the encrypted data is regulated by the gain factor γ in the embedding process. Therefore, the influence of gain factor on the simulation results is given in this section, which is plotted in Fig.8. Among them, the red curves represent the PSNR between the meaningful cipher images and the host images. Moreover, the blue curves represent the PSNR between the plain images and the decrypted images. It can be seen from the figure that as the value of the gain factor increases, the red curve gradually rises. While for the blue curve, it rises in the first stage and then falls. Additionally, in the case of weighing the quality of the decrypted image and the visual security of the cipher image, the optimal gain factor is different for different plain images.

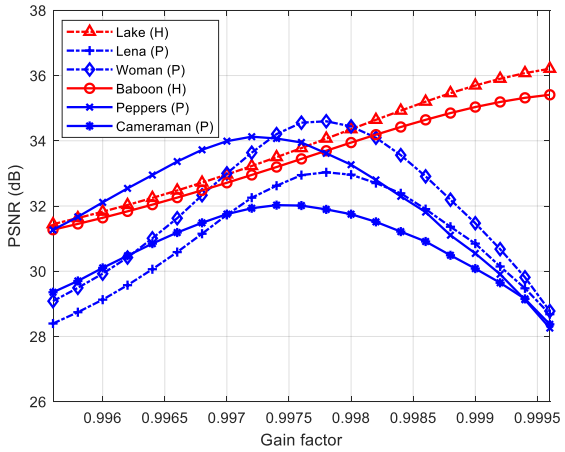


Fig.8 PSNR vs gain factor with different images.

5.4 Influence of embedding position on decryption quality

In the proposed DIVME algorithm, we introduce a FRFT-based embedding method, which can embed two plain images into the real and imaginary parts of complex matrix, respectively. Thus, this subsection will evaluate the influence of embedding position on decryption. First, two identical plain images are subjected to the proposed algorithm under the condition that $Ts1 = Ts2 = 35$ and other encryption parameters are consistent with those described in Section 5.1. Besides, the image Baboon is selected as the host image. The experimental results are plotted in Fig.9. As observed from Fig.9, the embedding position basically has no effect on the quality of decrypted image. Therefore, the plain images can be encrypted and freely embedded into different locations.

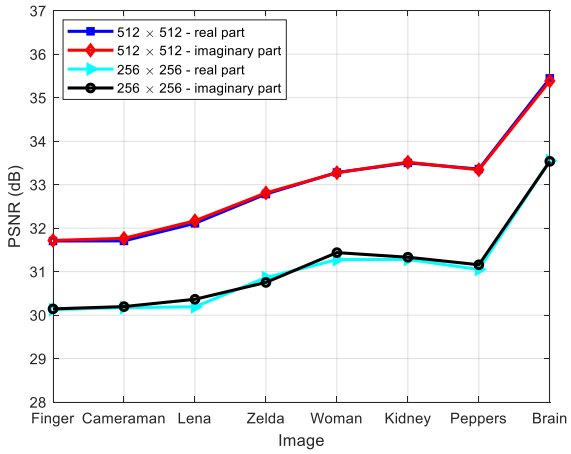


Fig.9 PSNR vs embedding position for different images with different resolutions.

5.5 Multiple attack analysis

5.5.1 Violent attack

The key space and key sensitivity together determine the ability of algorithm to resist violent attacks. In this paper, the secret keys mainly consist of the following three parts. *i.e*, (a) the (k_1, k_2, k_3, k_4) used for generating the initial values of two chaotic systems. (b) the (h_1, h_2) used for calculating the location of internal parameters in the alpha channel and (c) the rotation angle (α, β). Suppose that the step length of k_i $(i = 1, 2, 3, 4)$ and h_i $(i = 1, 2)$ are 10^{-14} and 10^{-10} respectively, while the step of rotation angle is 10^{-2}. Thus, the total key space can be calculated by Eq.(34).

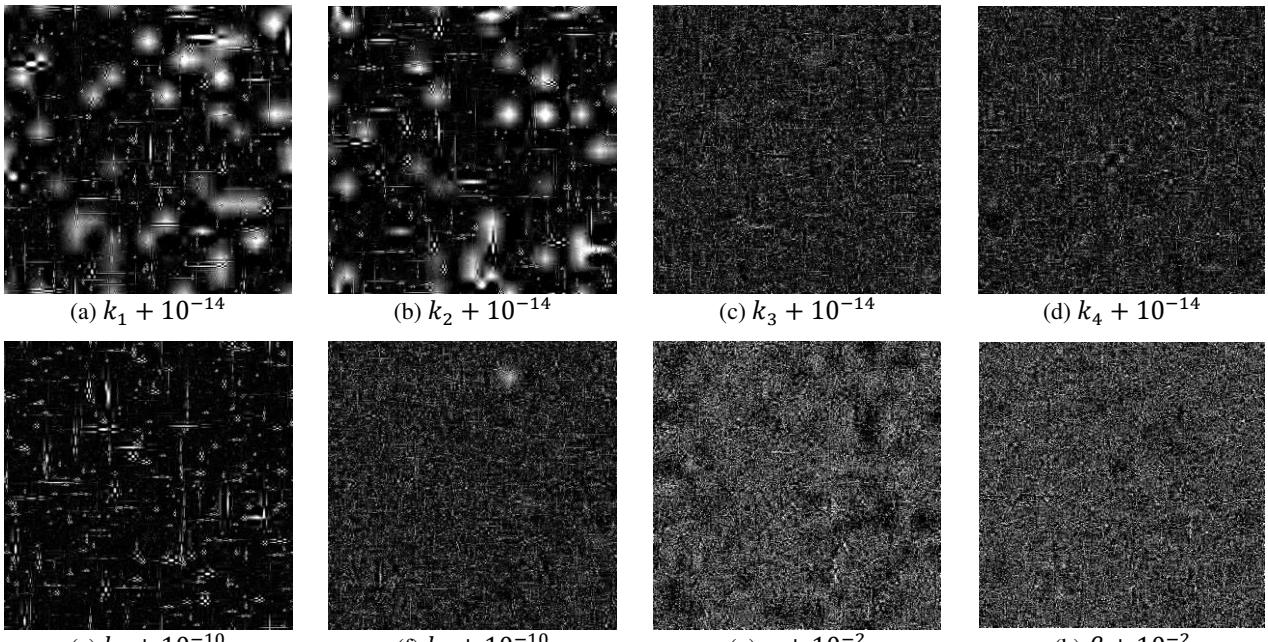
$$key_{total} = (10^{14})^4 \times (10^{10})^2 \times (2\pi \times 10^2)^2 = 4\pi^2 \times 10^{80} \tag{34}$$

Furthermore, the key space for the other algorithms is listed in Tab.2. It can be seen that compared with Ref.[42], the proposed DIVME algorithm has a larger key space.

Tab.2 Comparison of key space.

Algorithm	Ours	Ref.[19]	Ref.[31]	Ref.[33]	Ref.[43]
Key space	$4\pi^2 \times 10^{80}$	2.56×10^{59}	10^{56}	10^{56}	10^{75}

To qualitatively analyze the key sensitivity, the images Lena and Woman are subjected to the proposed algorithm. Then the modified secret keys by adding a slight change to one of the correct keys are used to decryption. Moreover, the decrypted images are illustrated in Fig.10. It can be clearly seen that when one of the correct keys is slightly changed, the decrypted image does not visually reveal any useful information of plain image, indicating that the proposed encryption scheme is sensitive to the keys. To sum up, our scheme is sufficiently resistant to violent attacks.



(a) $k_1 + 10^{-14}$ (b) $k_2 + 10^{-14}$ (c) $k_3 + 10^{-14}$ (d) $k_4 + 10^{-14}$

(e) $h_1 + 10^{-10}$ (f) $h_1 + 10^{-10}$ (g) $\alpha + 10^{-2}$ (h) $\beta + 10^{-2}$

Fig.10 Decrypted image “Lena” using incorrect keys.

5.5.2 Statistical attack

The pixel distribution of an image can be reflected by the histogram. Since the strong correlation between adjacent pixels in the natural image, its histogram always presents an uneven shape. However, for an effective visual image encryption algorithm, the histogram of the visually secure cipher image should be as consistent as possible with that of the corresponding host image. Next, the distance of histogram intersection [43] is introduced to measure the slight differences between host images and cipher images, which can be calculated by Eq.(35).

$$H(\mathrm{J},\mathrm{V}) = \frac{\sum_{i=1}^{2^L} \min(\mathrm{J}_i, \mathrm{V}_i)}{\sum_{i=1}^{2^L} \mathrm{V}_i} \tag{35}$$

Where (J, V) is a pair of histogram and L represents the bit depth of image. As shown in Eq.(34), when the histograms J and V are equal, the $H(\mathrm{J}, \mathrm{V})$ reaches its maximum value that is one. In the experiment, the plain images with different resolutions are encrypted and embedded into different host images. Then the obtained results are listed in Tab.3. It is observed that the distance of histogram intersection between the host images and the cipher images is close to one, indicating that the proposed DIVME algorithm has good visual security. Similarly, it can also be seen that the host images have little impact on the value of $H(\mathrm{J}, \mathrm{V})$.

Tab.3 The difference between the histograms of host images and cipher images.

Resolution	Plain image	Host image	Distance
256×256	Finger	Baboon	0.9311
	Cameraman	Goldhill	0.9266
		Peppers	0.9208
		Zelda	0.9472
512×512	Lena	Lake	0.9123
	Woman	Lung	0.9158
		Barbara	0.9387
		Baboon	0.9423

5.5.3 Noise attack

Considering that the meaningful cipher image transmitted over the channel will be inevitably affected by various noises, resulting in loss of partial ciphertext data, which increases the difficulty of recovering the plain image. We will perform several experiments to evaluate the anti-noise performance of the proposed DIVME algorithm in this subsection under the condition that the parameters *fv* and *gv* hidden in the alpha channel are not destroyed. First, the plain images 'Lena' and 'Woman' with resolution of 512×512 are encrypted and embedded into the host image 'Lake' via FRFT embedding. Then, multiple types of noise with normalized intensity of 0.0001%, 0.0005%, 0.001% and 0.005% are added to the meaningful cipher image respectively, including salt and pepper noise (SPN), speckled noise (SN) and Gaussian noise (GN). The resultant decrypted images are plotted in Fig.11-Fig.13. Besides, the values of PSNR between the decrypted image and the plain image under different noise attacks are listed in Tab.4.

As illustrated in Fig.11-Fig.13 and Tab.4, when the attack intensity varies from 0.0001% to 0.005%, the quality of the decrypted image drops significantly, and the maximum drop reach 4.787 dB, but we can still visually identify the secret information carried by the decrypted image. In another aspect, it can be seen from the experimental data that in the case of the same noise attack intensity, GN has the greatest impact on our proposed scheme, while SPN and SN have the least impact. In general, the proposed DIVME scheme has a good ability to resist noise interference.

Cipher image

(a) 0.0001% SPN (b) 0.0005% SPN (c) 0.001% SPN (d) 0.005% SPN

(e) 0.0001% SPN (f) 0.0005% SPN (g) 0.001% SPN (h) 0.005% SPN

(i) 0.0001% SPN (j) 0.0005% SPN (k) 0.001% SPN (l) 0.005% SPN

Fig.11 Simulation results under SPN with different intensities.

(a) 0.0001% SN (b) 0.0005% SN (c) 0.001% SN (d) 0.005% SN

(e) 0.0001% SN (f) 0.0005% SN (g) 0.001% SN (h) 0.005% SN

(i) 0.0001% SN (j) 0.0005% SN (k) 0.001% SN (l) 0.005% SN

Fig.12 Simulation results under SN with different intensities.

(a) 0.0001% GN (b) 0.0005% GN (c) 0.001% GN (d) 0.005% GN

(e) 0.0001% GN (f) 0.0005% GN (g) 0.001% GN (h) 0.005% GN

(i) 0.0001% GN (j) 0.0005% GN (k) 0.001% GN (l) 0.005% GN

Fig.13 Simulation results under GN with different intensities.

Tab.4 PSNR values under different noise attacks.

Noise type	Intensity			
	0.0001%	0.0005%	0.001%	0.005%
SPN	$Psnr_1 = 32.9598$	$Psnr_1 = 32.9598$	$Psnr_1 = 31.1012$	$Psnr_1 = 29.8191$
	$Psnr_2 = 34.4402$	$Psnr_2 = 34.4402$	$Psnr_2 = 31.9757$	$Psnr_2 = 30.2442$
SN	$Psnr_1 = 32.9598$	$Psnr_1 = 32.1540$	$Psnr_1 = 31.6417$	$Psnr_1 = 30.1879$
	$Psnr_2 = 34.4402$	$Psnr_2 = 33.3408$	$Psnr_2 = 32.5616$	$Psnr_2 = 30.7421$
GN	$Psnr_1 = 32.5458$	$Psnr_1 = 30.9543$	$Psnr_1 = 30.3381$	$Psnr_1 = 28.8122$
	$Psnr_2 = 33.8099$	$Psnr_2 = 31.6741$	$Psnr_2 = 30.8617$	$Psnr_2 = 29.0229$

5.5.4 Cropping attack

The ability of the proposed DIVME algorithm to resist cropping attacks is also evaluated in this subsection. Similarly, it is assumed that the parameters *fv* and *gv* in the meaningful cipher image are not corrupted. Then, different positions of the cipher images are cut with sizes of 128×128, 180×180 and 256×256, which are drawn in the first row of Fig.14. And the corresponding decrypted images are depicted in the second and third row of Fig.14.

It can be seen from the figure that the position of the cropping blocks has basically no impact on the visual quality of the decrypted image in our scheme. Besides, as the size of the cropping block increases, the quality of the decrypted image decreases. However, when a quarter of the cipher image data is lost, we can still find the information carried by the plain image in the decrypted image visually, and the PSNR of the decrypted images are about 26.7 dB and 26.8 dB respectively, indicating that our DIVME scheme can withstand the cropping attacks to a certain extent, provided that the eigenvalues are not lost.

(a) $1/16$ data loss (b) $1/8$ data loss (c) $1/8$ data loss (d) $1/4$ data loss (e) $1/4$ data loss

(f) $Psnr = 27.9681$ (g) $Psnr = 27.3165$ (h) $Psnr = 27.3238$ (i) $Psnr = 26.7010$ (j) $Psnr = 26.6695$

(k) $Psnr = 28.1599$ (l) $Psnr = 27.4223$ (m) $Psnr = 27.4153$ (n) $Psnr = 26.8275$ (o) $Psnr = 26.8474$

Fig.14 Decrypted results of the cipher image (512×512) suffering from different cropping attacks.

5.5.5 Differential attack

Differential attack is the most common attack method used by attackers. It refers to attempting to explore the relationship between the plain image and the cipher image by analyzing the cipher images produced by two plain images which differ by only one pixel, and then crack the secret information carried in other cipher images without using the key. The NPCR and UACI [44-45] are adopted to quantitatively measure the ability of our scheme against the differential attack in this section. And their calculation equations are displayed in Eq.(36) and Eq.(37).

$$Npcr = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|\mathrm{Sign}(\mathbf{I}_1^{i,j} - \mathbf{I}_2^{i,j})\right| \times 100\% \tag{36}$$

$$Uaci = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\left|\mathbf{I}_1^{i,j} - \mathbf{I}_2^{i,j}\right|}{255} \times 100\% \tag{37}$$

In Eq.(36), the symbol Sign stands for the sign function. When $\mathbf{I}_1^{i,j}$ is equal to $\mathbf{I}_2^{i,j}$, the value of the sign function is 0, otherwise it is ± 1. In our scheme, the resulting cipher image is visually similar to the host image. Thus, the smaller the values of NPCR and UACI are, the more difficult it is to find the relationship between the cipher image and the plain image. Tab.5 gives the NPCR and UACI values for plain images with different resolutions. As illustrated in Tab.5, the values of NPCR and UACI are very close to 0, indicating that the proposed DIVME algorithm has strong resistance to differential attacks.

Tab.5 The NPCR and UACI values for different resolutions of images.

Resolution	Plain image	Host image	NPCR	UACI
256×256	Finger Cameraman	Baboon	1.5259×10^{-3} %	5.9838×10^{-6} %
	Peppers Zelda	Goldhill	1.5259×10^{-3} %	5.9838×10^{-6} %
512×512	Lena Woman	Lake	1.1444×10^{-3} %	4.4879×10^{-6} %
	Kidney Brain	Lung	0 %	0 %

5.5.6 Known-plaintext and chosen-plaintext attacks

Up to now, many image encryption algorithms have been attacked by chosen-plaintext and known-plaintext, such as

Ref.[46-48]. The main reason is that different plain images correspond to the same key stream in encryption. However, in this paper, the eigenvalues of the two plain images are utilized to control the generated key stream. First, the initial values of the QCNN are determined by the eigenvalue fv. Then, it is iterated and sorted to fabricate two sets of index sequences. One set of sequences is applied to control the Fisher-Yates confusion performed on the image. And the remaining set of sequences is used to randomly embed the encrypted data into the host image. In other aspects, the eigenvalue gv is adopted to control the IHM to generate the key-controlled partial Hadamard matrix, which is used to compress two plain images in parallel. In short, since the proposed DIVME algorithm can realize "One plain image corresponds to one key", it can withstand known-plaintext and chosen-plaintext attacks.

Additionally, in order not to violate the basic principles of symmetric cryptosystems, we propose to embed the eigenvalues into the meaningful cipher image and transmit it to the decoders. Moreover, the scheme of hiding the eigenvalues has two characteristics. First, the location of the eigenvalues is controlled by the key. Secondly, the eigenvalues are embedded in the alpha channel of the visual cipher image to reduce the probability of being attacked.

Tab.6 Encryption time for plain images with different resolutions (Unit: s).

Resolution	Plain image	Pretreatment	Pre-encryption	FRFT embedding	Total
256 × 256	Cameraman Goldhill	0.005864	0.028206	0.087978	0.122048
	Finger Peppers	0.007858	0.021990	0.081224	0.111072
	Average	0.006861	0.025098	0.084601	
512 × 512	Lena Woman	0.024722	0.093751	1.004689	1.123162
	Kidney Brain	0.025451	0.089174	0.974852	1.089477
	Average	0.025087	0.091463	0.989771	

Tab.7 Decryption time for plain images with different resolutions (Unit: s).

Resolution	Plain image	Pretreatment	Extraction	Reconstruction	Total
256 × 256	Cameraman Goldhill	0.010422	0.037255	0.984013	1.031690
	Finger Peppers	0.008713	0.050944	0.944407	1.004064
	Average	0.009568	0.044100	0.964210	
512 × 512	Lena Woman	0.029605	0.548270	5.285881	5.863756
	Kidney Brain	0.029082	0.590665	5.451860	6.071607
	Average	0.029344	0.569468	5.368871	

5.6 Running efficiency analysis

Except for considering the security of algorithm, running efficiency is also an indispensable indicator to evaluate the performance of algorithm in practical applications. The execution times required for encryption and decryption of images with different resolutions are listed in Tab.6-Tab.7. In these two tables, 'Pretreatment' represents the phase that generates the initial values of the chaotic systems, and 'Reconstruction' stands for the process of recovering two natural images from the encrypted data through the SL_0 reconstruction algorithm and the inverse Fisher-Yates scrambling.

As can be seen, embedding and reconstructing the encrypted data account for a large proportion of the total encryption and decryption time. Such as for the plain image with size of 512×512, the FRFT embedding phase takes up about 89.465% of the total time in encryption. Moreover, the reconstruction phase occupies around 89.966% of the total decryption time. In another aspect, when the resolution of the plain image changes from 256×256 to 512×512, the time spent in the encryption and decryption process increases about 10 times and 6 times, respectively. Thus, it is suggested to divide the large-scale plain image into several small images and then perform encryption and embedding operations in parallel to shorten the execution time.

6 Comparison with the existing work

In order to highlight the proposed DIVME algorithm, first, we summarize the characteristics of the existing visually meaningful image encryption algorithm, and compare them with ours. The comparison results are displayed in Tab.8. Besides, part of them are contrasted with our proposed algorithm from the following three aspects: visual security, anti-noise performance and running efficiency, as shown in Tab.9-Tab.11. Note that for the sake of fairness, the experimental data of Ref.[19,32,43] are obtained from their source articles or the related articles, and N/A means that the value is not provided. It can be seen from the comparison results that our DIVME algorithm has better anti-noise performance and higher running efficiency, compared with that in Ref.[32,43]. As for the visual security, the proposed scheme in this paper is comparable to the Ref.[43], but both are significantly better than that of Ref.[32].

Tab.8 Comparison of the characteristics for different algorithms.

No	Other algorithms	Our
1	The plain image with resolution of $m \times n$ is pre-encrypted and embedded into a host image with resolution of $2m \times 2n$ to obtain a meaningful cipher image, thus increasing the cost of storage and transmission, such as Ref.[29,33].	Before the embedding phase, the compressed sensing is adopted to compress the plain image to a quarter of its original resolution. Thus, the visual cipher image has the same resolution as the plain image in our scheme.
2	Only one plain image can be encrypted at a time, such as Ref.[49-52]. The encryption efficiency and transmission efficiency are limited.	Two plain images can be simultaneously encrypted and embedded into the host image with the same resolution as the plain image via the FRFT embedding method in this paper. The transmission efficiency has been improved, and there is no need to provide additional transmission costs.
3	There are both quantization error and truncation error in Ref.[50-52]. Because of the accumulation of errors, the visual quality of the decrypted image is reduced.	In the proposed DIVME algorithm, to reduce the influence of errors, the encrypted data is directly embedded into the host image under the control of gain factor, without quantization operation.
4	The existing embedding methods, such as LSB [31], SVD [19, 33], IWT [29,30], RCT [50], DCT [53] and so on, cannot protect the hidden ciphertext.	In the proposed FRFT-based embedding method, the rotation angles of the fractional Fourier transform can be used as the keys to improve the security of the hidden ciphertext.
5	Part of the existing visually meaningful image encryption algorithms are vulnerable to the chosen-plaintext and known-plaintext attacks, since in their schemes, different plain images correspond to the same key stream.	The characteristic information of the two plain images is utilized to generate the key stream in encryption, so that different plain images correspond to different key streams. Additionally, in order not to violate the basic principles of the symmetric encryption cryptosystem, the eigenvalues are embedded in the alpha channel of the meaningful cipher image under the control of the key.

Tab.9 Comparisons of the PSNR and MSSIM between the cipher and host images in different encryption schemes.

Plain image	Host image	Ref.[32]		Ref.[43]		Ref.[19]		Ours	
		PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM
Lena	Peppers	18.5136	0.6726	32.3513	0.9257	31.7986	0.9903	36.0020	0.9963
Brain	Cameraman	24.8700	0.6488	34.8967	0.9381	31.1582	0.9725	35.3293	0.9985
Jet	Baboon	23.3967	0.6991	37.8967	0.9833	32.6033	0.9955	34.2920	0.9984
Barbara	Bridge	25.2321	0.7337	35.5629	0.9783	31.7397	0.9946	34.1259	0.9982
Average		**23.0031**	**0.6886**	**35.1769**	**0.9564**	**31.8250**	**0.9882**	**34.9373**	**0.9979**

* In order to compare with the single-image visually meaningful encryption algorithm, in this experiment, two identical plain images are placed in the proposed DIVME algorithm.

Tab.10 Comparison of the resistance capability of noise attacks.

Nosie Type	PSNR								
	Noise Intensity = 0.0001%			Noise Intensity = 0.0003%			Noise Intensity = 0.0005%		
	Ref.[43]	Ref.[19]	Our	Ref.[43]	Ref.[19]	Our	Ref.[43]	Ref.[19]	Our
GN	14.75	25.13	33.49	9.35	19.41	32.31	8.54	17.78	31.84
SN	33.44	31.56	33.97	22.50	27.68	33.97	17.30	24.39	33.97
SPN	33.44	31.56	33.97	33.44	30.22	33.97	33.44	30.02	33.92

* In the above experiment, the plain image and host image are selected as Jet and Coldhill, respectively.

Tab.11 Comparisons results of encryption and decryption times in different encryption schemes (Unit: f/s).

Resolution	Ref.[32]		Ref.[43]		Ref.[19]		Our	
	Encryption	Decryption	Encryption	Decryption	Encryption	Decryption	Encryption	Decryption
256×256	0.4049	1.4724	0.1533	2.2679	0.3546	0.5192	0.0583	0.5089
512×512	0.9279	10.5830	N/A	N/A	0.6185	2.2219	0.5532	2.9838

7 Conclusions

This paper introduces an efficient DIVME algorithm combined with QCNN, CS and FRFT-based embedding method, which can simultaneously realize the image data security and appearance security. Besides, for withstanding the plaintext attacks, the eigenvalues of the plain images are adopted to generate the key streams. And then these eigenvalues are embedded into the alpha channel of the meaningful cipher image to reduce the probability of being destroyed. Finally, a series of security analysis indicate that the proposed DIVME algorithm not only has high visual security and decryption quality, but also can resist the diversified attacks, such as violent attack, noise attack, chosen-plaintext attack and so on. In the following work, we will devote ourselves to further improving the sparsity of plain images to attain the higher image compression rate.

Conflicts of Interest

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China [Grant No.61701043, 41874140], the Shaanxi Province Science and Technology Program [Grant No.2020JM-220, 2020JQ-351], the Fundamental Research Funds for the Central Universities of China [Grant No.300102240205], the Natural Science Foundation of Fujian Province [Grant No.2020J05169] and the Natural Science Foundation of Heilongjiang Province [Grant No.F2018022].

References

[1] Irani, B., Ayubi, P., Jabalkandi, F., Valandar, M., Barani, M.: Digital image scrambling based on a new one-dimensional coupled Sine map. Nonlinear Dyn. 97, 2693-2721 (2019).

[2] Sun, Y., Zhang, H., Wang, X., Wang, X., Yan, P.: 2D Non-adjacent coupled map lattice with q and its applications in image encryption. Appl. Math. Comput. 373,125039 (2020).

[3] Candes, E., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory 52, 489-509 (2006).

[4] Donoho, D.: Compressed sensing. IEEE Trans. Inf. Theory 52, 1289-1306 (2006).

[5] Rachlin, Y., Baron, D.: The secrecy of compressed sensing measurements. in: Proceedings of the Allerton Conference on Communication, Control and Computing, 813-817 (2008).

[6] Zhang, L., Wong, K., Zhang, Y., Zhou, J.: Bi-level Protected Compressive Sampling. IEEE T. Multimedia 18(9), 1720-1732 (2016).

[7] Gong, L., Qiu, K., Deng, C., Zhou, N.: An image compression and encryption algorithm based on chaotic system and compressive sensing. Opt. Laser Technol. 115, 257-267 (2019).

[8] Hua, Z., Zhou, Y.: Exponential chaotic model for generating robust chaos. IEEE Trans. Syst. Man Cybern. -Syst. DOI: 10.1109/TSMC.2019.2932616.

[9] Cao, W., Mao, Y., Zhou, Y.: Designing a 2D infinite collapse map for image encryption. Signal Process. 171, 107457 (2020).

[10] Farah, M., Farah, A., Farah, T.: An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. Nonlinear Dyn. 99, 3041-3064 (2020).

[11] Wen, W., Wei, K., Zhang, Y., Fang, Y., Li, M.: Colour light field image encryption based on DNA sequences and chaotic systems. Nonlinear Dyn. 99, 1587-1600 (2020).

[12] Wang, X., Guan, N.: A novel chaotic image encryption algorithm based on extended Zigzag confusion and RNA operation. Opt. Laser Technol. 131, 106366 (2020).

[13] Naskar, P, Bhattacharyya, S., Nandy, D., Chaudhuri, A.: A robust image encryption scheme using chaotic tent map and cellular automata. Nonlinear Dyn. 100, 2877-2898 (2020).

[14] Mondal, B., Singh, S., Kumar, P.: A secure image encryption scheme based on cellular automata and chaotic skew tent map. J. Inf. Secur. Appl. 45, 117-130 (2019).

[15] Wang, X., Li. Z.: A color image encryption algorithm based on Hopfield chaotic neural network. Opt. Laser. Eng. 115, 107-118 (2019).

[16] Yang, F., Mou, J., Cao, Y., Chu, R.: An image encryption algorithm based on BP neural network and hyperchaotic system. China Commun. 17(5), 21-28 (2020).

[17] Huang, R., Rhee, K., Uchida, S.: A parallel image encryption method based on compressive sensing. Multimed. Tools Appl. 72, 71-93 (2014).

[18] Zhou, N., Zhang, A., Wu, J., Pei, D., Yang, Y.: Novel hybrid image compression-encryption algorithm based on compressive sensing. Optik 125, 5075-5080 (2014).

[19] Zhu, L., Song, H., Zhang, X., Yan, M., Zhang, T., Wang, X., Xu, J.: A robust meaningful image encryption scheme based on block compressive sensing and SVD embedding. Signal Process. 175, 107629 (2020).

[20] Zhao, H., Ye, H., Wang, R.: The construction of measurement matrices based on block weighing matrix in compressed sensing. Signal Process. 123, 64-74 (2016).

[21] Souyah, A., Faraoun, K.: Fast and efficient randomized encryption scheme for digital images based on quadtree decomposition and reversible memory cellular automata. Nonlinear Dyn. 84(2), 715-732 (2016).

[22] Zhou, N., Pan, S., Cheng, S., Zhou, Z.: Image compression-encryption scheme based on hyper-chaotic system and 2D compressive

sensing. Opt. Laser Technol. 82, 121-133 (2016).

[23] Gong, L., Qiu, K., Deng, C., Zhou, N.: An image compression and encryption algorithm based on chaotic system and compressive sensing. Opt. Laser Technol. 115, 257-267 (2019).

[24] Zhu, L., Song, H., Zhang, X., Yan, M., Zhang, L. Yan, T.: A novel image encryption scheme based on nonuniform sampling in block compressive sensing. IEEE Access, 7, 22161-22174 (2019).

[25] Hu, G., Xiao, D., Wang, Y., Xiang, T., Zhou, Q.: Securing image information using double random phase encoding and parallel compressive sensing with updated sampling processes. Opt. Laser. Eng. 98, 123-133 (2017).

[26] Hu, G., Xiao, D., Wang, Y., Xiang, T.: An image coding scheme using parallel compressive sensing for simultaneous compression-encryption applications. J. Vis. Commun. Image Represent. 44, 116-127 (2017).

[27] Ponuma, R., Amutha, R.: Compressive sensing based image compression-encryption using Novel 1D-Chaotic map, Multimed. Tools. Appl. 77 19209-19234 (2017).

[28] Gan, Z., Chai, X., Zhang, J., Zhang, Y., Chen, Y.: An effective image compression-encryption scheme based on compressive sensing (CS) and game of life (GOL). Neural Comput. Applic. 32, 14113-14141 (2020).

[29] Bao L., Zhou Y.: Image encryption: generating visually meaningful encrypted images. Inf. Sci. 324, 197-207 (2015).

[30] Khan, J., Boulila, W., Ahmad, J., Rubaiee, S., Rehman, A., Alroobaea, R., Buchanan, W.: DNA and plaintext dependent chaotic visual selective image encryption. IEEE Access 8, 159732-159744 (2020).

[31] Chai, X., Wu, H., Gan, Z., Zhang, Y., Chen, Y., Nixon, K.: An efficient visually meaningful image compression and encryption scheme based on compressive sensing and dynamic LSB embedding. Opt. Laser. Eng. 124, 105837 (2020).

[32] Chai, X., Gan, Z., Chen, Y., Zhang, Y.: A visually secure image encryption scheme based on compressive sensing. Signal Process. 134, 35-51 (2017).

[33] Ye, G., Pan, C., Dong, Y., Shi, Y., Huang, X.: Image encryption and hiding algorithm based on compressive sensing and random numbers insertion. Signal Process. 172, 107563 (2020).

[34] Ye, G., Pan, C., Dong, Y., Jiao, K., Huang, X.: A novel multi-image visually meaningful encryption algorithm based on compressive sensing and Schur decomposition. Trans. Emerg. Telecommun. Technol. DOI: 10.1002/ett.4071.

[35] Li, J., Di, X., Liu, X., Chen, X.: Image encryption based on quantum-CNN hyperchaos system and Anamorphic Fractional Fourier Transform. In 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, 1-6 (2017).

[36] Zhao, H., Xie, S., Zhang, J., Wu, T.: Fast image encryption algorithm based on improved Henon map. Application Research of Computers 37(12), 3726-3730 (2020).

[37] Baraniuk, R.: Compressive sensing. IEEE Signal Process. Mag. 24, 118-121 (2007).

[38] Wang, X., Su, Y.: Color image encryption based on chaotic compressed sensing and two-dimensional fractional Fourier transform. Sci. Rep. 10, 18556 (2020).

[39] Elhoseny, H., Faragallah, O., Ahmed, H., Kazemian, H., El-sayed, H., El-Samie, F., The effect of fractional Fourier transform angle in encryption quality for digital images. Optik, 127, 315-319 (2016).

[40] Parah, S., Sheikh, J., Loan, N., Ahad, F., Bhat, G.: Utilizing neighborhood coefficient correlation: a new image watermarking technique robust to singular and hybrid attacks. Multidim. Syst. Sign. Process. 29, 1095-1117 (2018).

[41] Yang, Y., Wang, B., Yang, Y., Zhou, Y., Shi, W.: Dual embedding model: a new framework for visually meaningful image encryption. Multimed. Tools Appl. DOI: 10.1007/s11042-020-10149-4.

[42] Liu, L., Jiang, D., Wang, X., Zhang, L., Rong, X.: A dynamic triple-image encryption scheme based on chaos, S-box and image

compressing. IEEE Access 8, 210382-210399 (2020).

[43] Wang, H., Xiao, D., Li, M., Xiang, Y., Li, X.: A visually secure image encryption scheme based on parallel compressive sensing. Signal Process. 155, 218-232 (2019).

[44] Wang, X., Wang, Y., Zhu, X., Luo, C.: A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level. Opt. Laser. Eng. 125, 105851 (2020).

[45] Alghafis, A., Firdousi, F., Khan, M., Batool, S., Amin, M.: An efficient image encryption scheme based on chaotic and Deoxyribonucleic acid sequencing. Math. Comput. Simul. 177, 441-466 (2020).

[46] Pak, C., Huang, L.: A new color image encryption using combination of the 1D chaotic map. Signal Process. 138, 129-137 (2017).

[47] Li, Z., Peng, C., Li, L., Zhu, X.: A novel plaintext-related image encryption scheme using hyper-chaotic system. Nonlinear Dyn. 94, 1319-1333 (2018).

[48] Ye, G., Pan, C., Huang, X., Zhao, Z., He, J.: A chaotic image encryption algorithm based on information entropy. Int. J. Bifurcation Chaos 28(1), 1850010 (2018).

[49] Yang, Y., Zou, L., Zhou, Y., Shi, W.: Visually meaningful encryption for color images by using Qi hyperchaotic system and singular value decomposition in YCbCr color space. Optik 213, 164422 (2020).

[50] Ping, P., Fu, J., Mao, Y., Xu, F., Gao, J.: Meaningful encryption: Generating visually meaningful encrypted images by compressive sensing and reversible color transformation. IEEE Access 7, 170168-170184 (2019).

[51] Wen, W., Hong, Y., Fang, Y., Li, M., Li, M.: A visually secure image encryption scheme based on semi-tensor product compressed sensing. Signal Process. 173, 107580 (2020).

[52] Musanna, F., Dangwal, D., Kumar, S.: A novel chaos-based approach in conjunction with MR-SVD and pairing function for generating visually meaningful cipher images. Multimed. Tools Appl. 79, 25115-25142 (2020).

[53] Cheng, W., Huang, J., Liu H.: A 3D-DCT-based information hiding algorithm for color images, Acta Automatica sinica 29, 258-266 (2003).

Figures

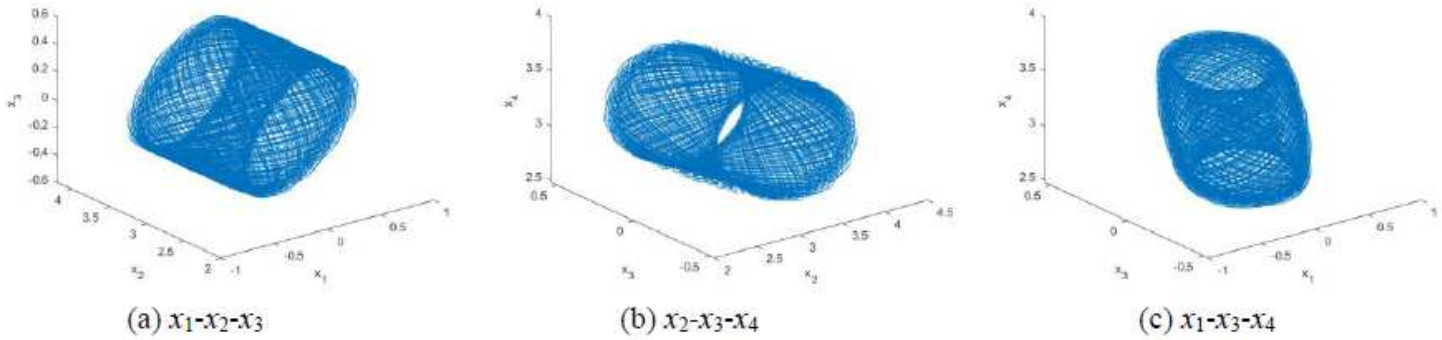


Figure 1

The hyperchaotic trajectories of this quantum cellular neural network.

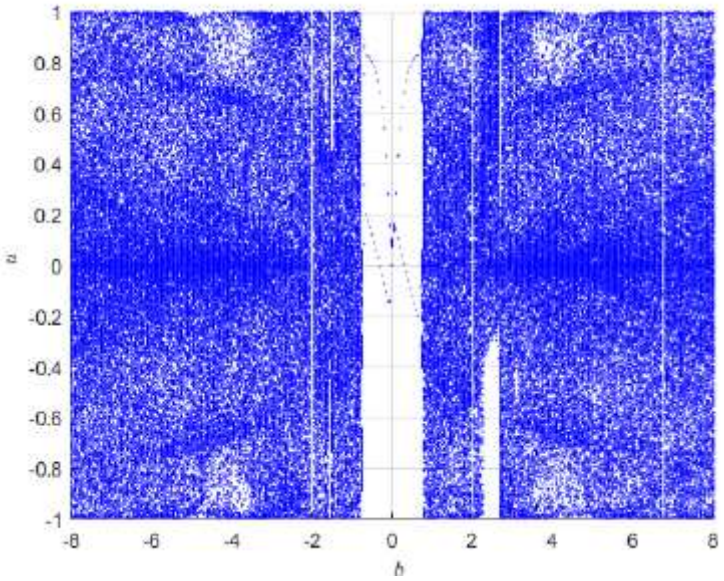


Figure 2

The bifurcation diagram of improved Henon map.

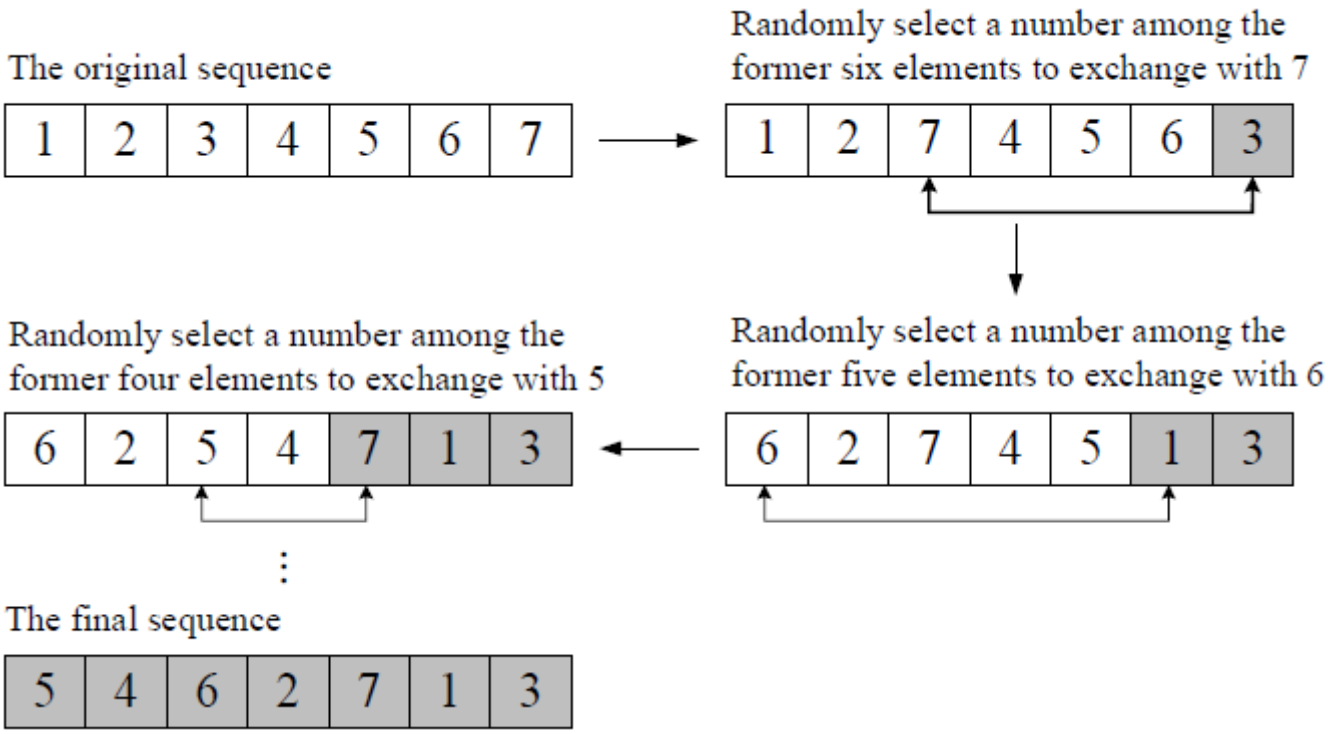


Figure 3

The scrambling process of Fisher-Yates algorithm.

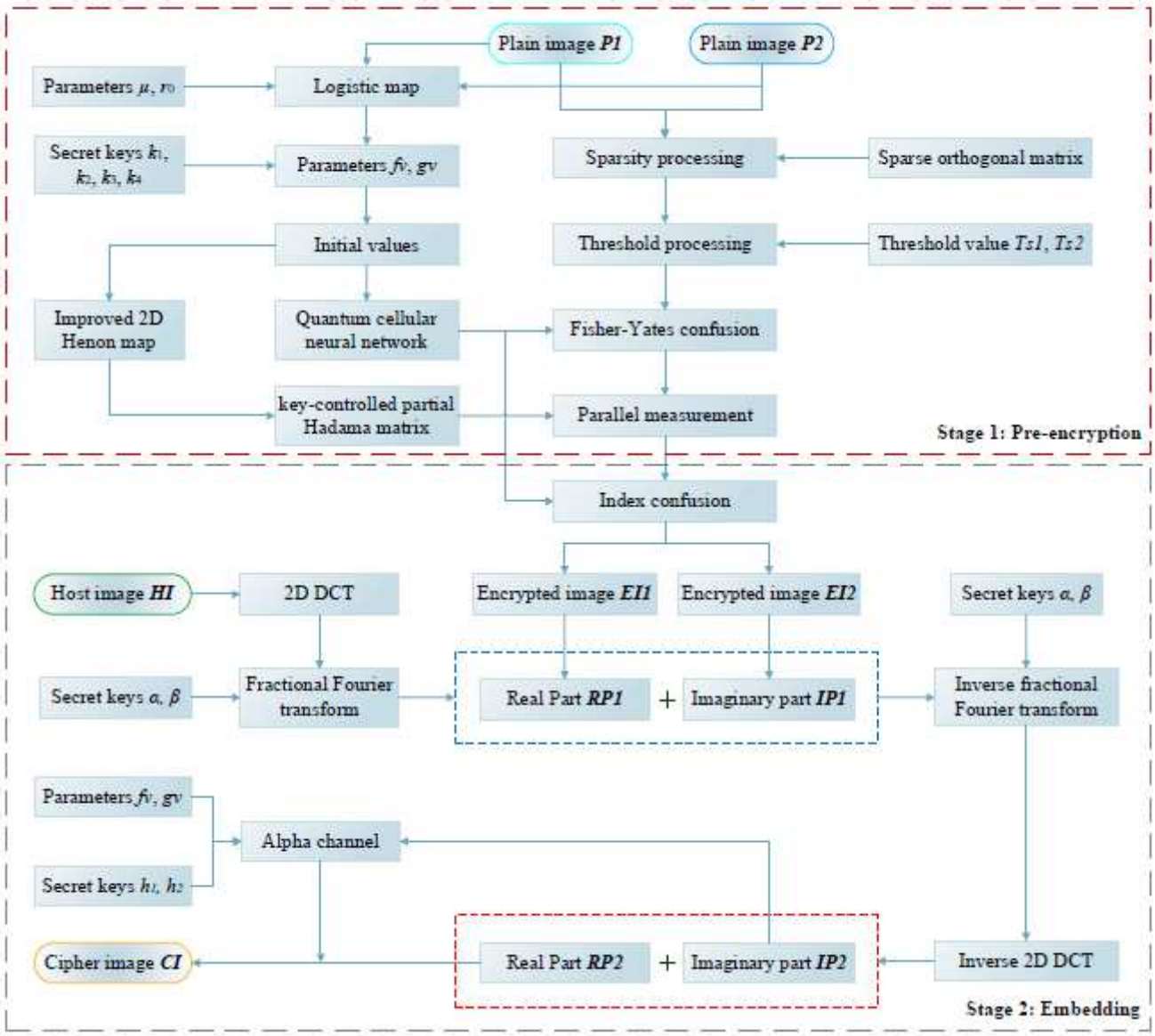


Figure 4

The schematic of the proposed DIVME algorithm.

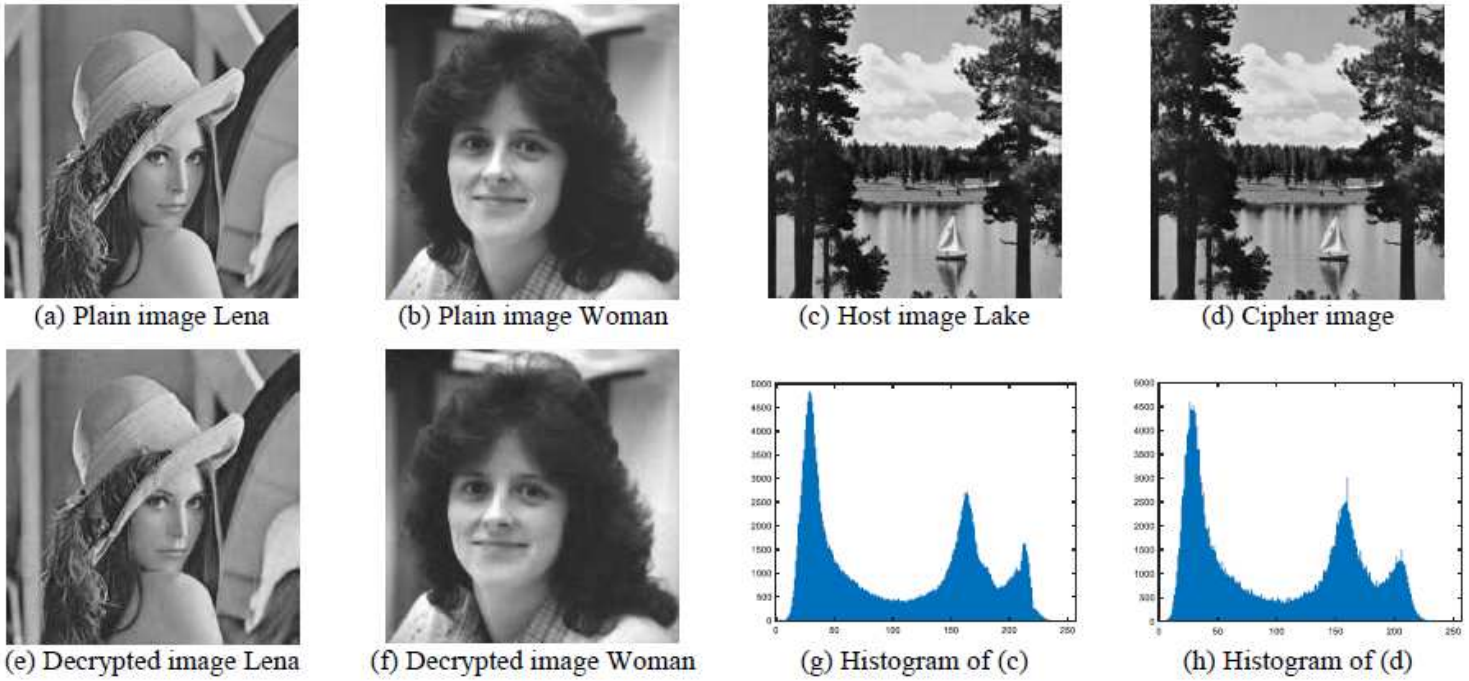


Figure 5

Simulation results with Lena and Woman as plain images (resolution 512 × 512).

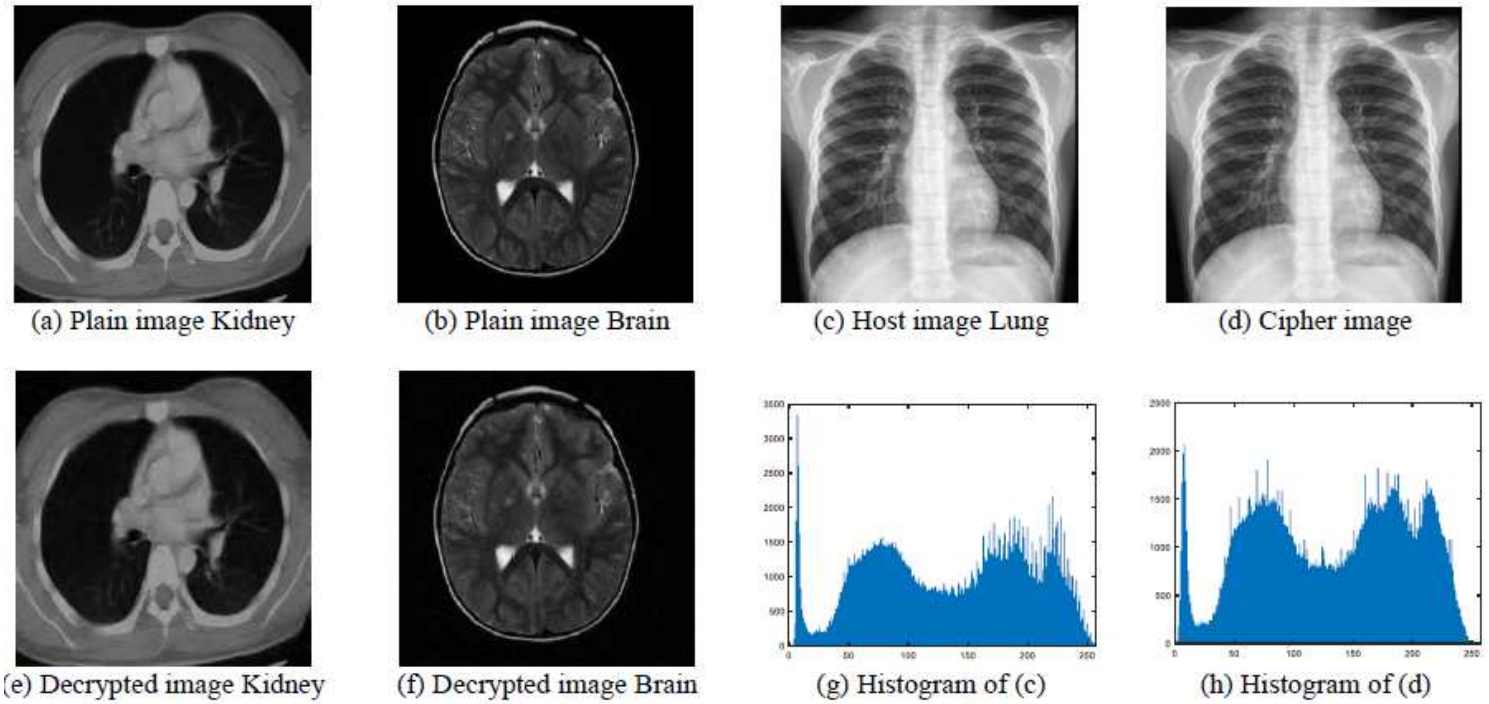


Figure 6

Simulation results with Kidney and Brain as plain images (resolution 512 × 512).

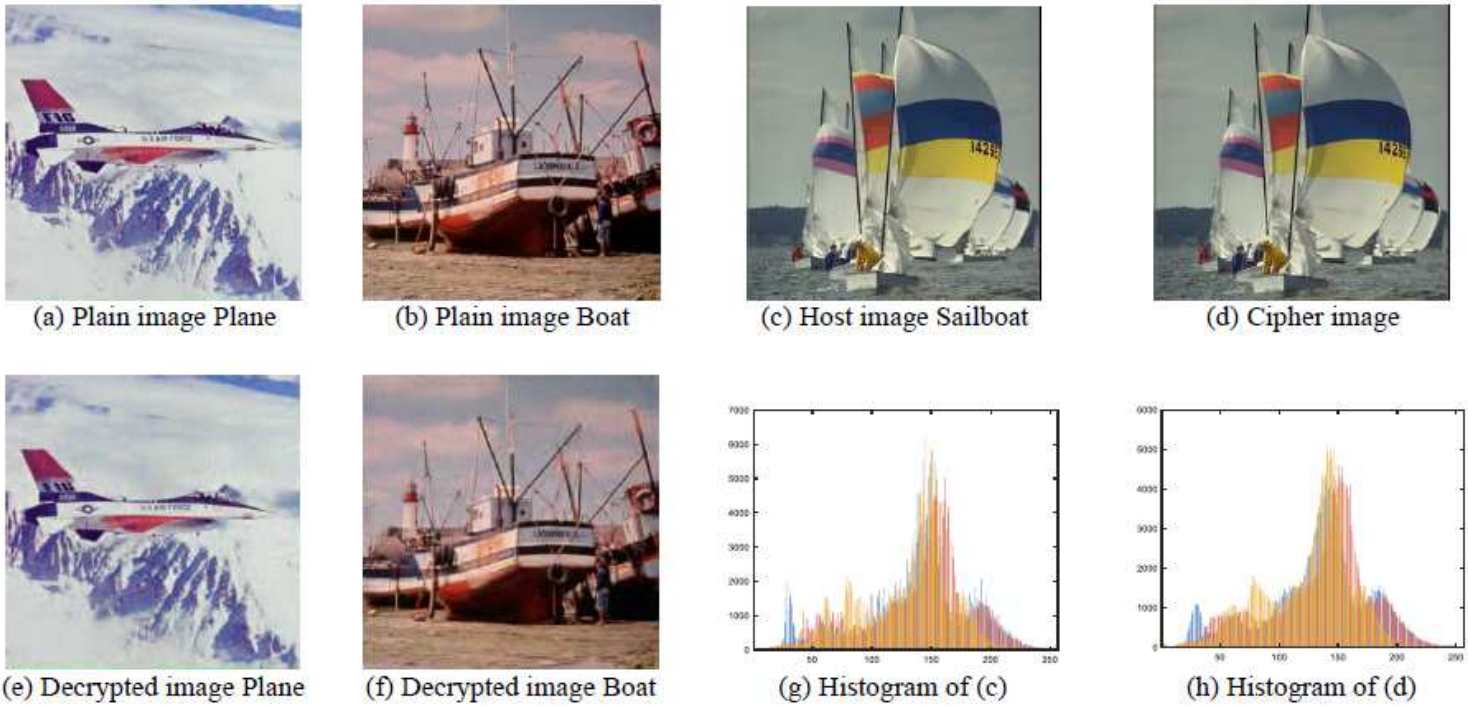


Figure 7

Simulation results with Plane and Boat as plain images (resolution 512 × 512 × 3).

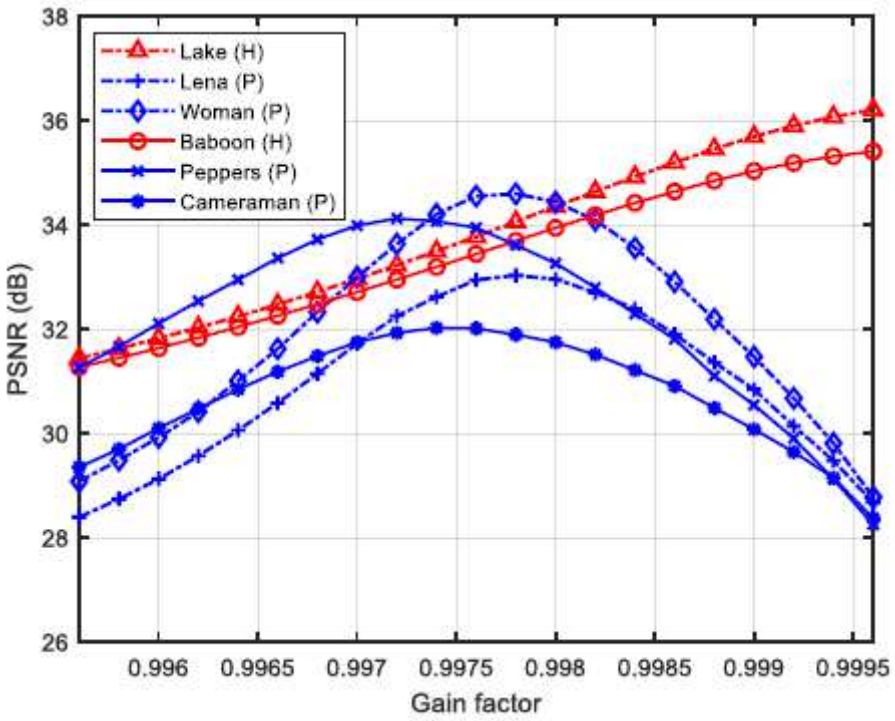


Figure 8

PSNR vs gain factor with different images.

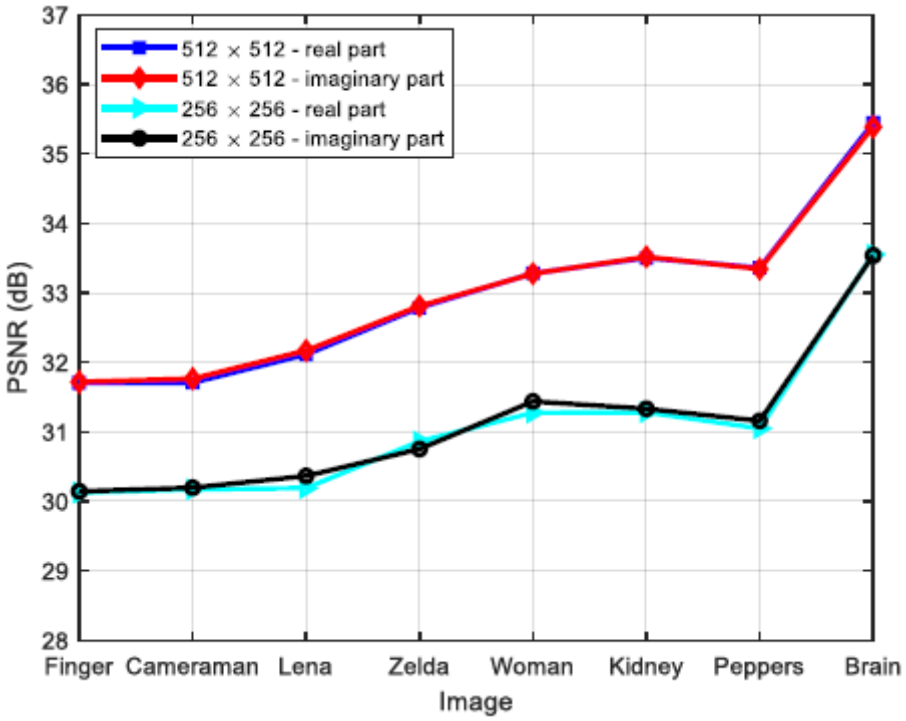


Figure 9

PSNR vs embedding position for different images with different resolutions.

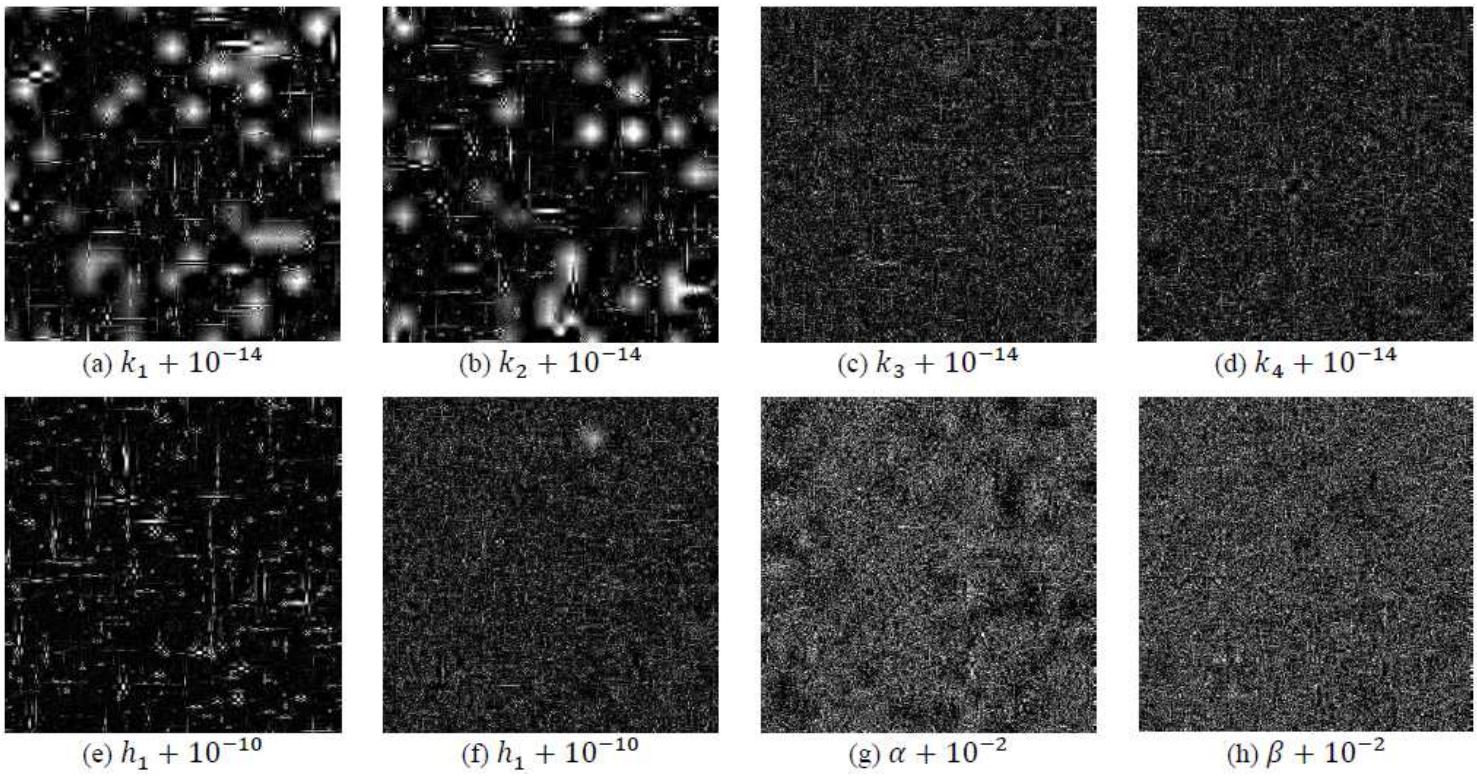


Figure 10

Decrypted image “Lena” using incorrect keys.

Figure 11

Simulation results under SPN with different intensities.

Figure 12

Simulation results under SN with different intensities.

Figure 13

Simulation results under GN with different intensities.

Figure 14

Decrypted results of the cipher image (512 × 512) suffering from different cropping attacks.