Journal Pre-proof

Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing

Zhongyun Hua, Kuiyuan Zhang, Yuanman Li, Yicong Zhou

PII: S0165-1684(21)00037-2
DOI: https://doi.org/10.1016/j.sigpro.2021.107998
Reference: SIGPRO 107998

To appear in: *Signal Processing*

Received date: 28 October 2020
Revised date: 28 December 2020
Accepted date: 19 January 2021

Please cite this article as: Zhongyun Hua, Kuiyuan Zhang, Yuanman Li, Yicong Zhou, Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing, *Signal Processing* (2021), doi: https://doi.org/10.1016/j.sigpro.2021.107998

- To improve the sparsification performance, we propose the adaptive-thresholding sparsification strategy, which utilizes SWT and column-based adaptive thresholding (CBAT).

- To improve the efficiency, we generate a random-order Bernoulli random matrix for each column in the data sampling and develop a new PCS technique.

- To reduce the data loss of carrier image, we introduce the matrix encoding technique to embed the secret image to the carrier image.

- Simulation and comparison results demonstrate that our proposed scheme is more efficient, and can achieve higher quality of the reconstructed and cipher images than some newly developed schemes.

Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing

Zhongyun Hua[a], Kuiyuan Zhang[a], Yuanman Li[b], Yicong Zhou[c]

[a]*School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China*
[b]*College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China*
[c]*Computer and Information Science, University of Macau, Macau 99907, China*

Abstract

Recently, some visually secure image encryption schemes using compressive sensing (CS) have been developed to protect images with visual security, where the images are first encrypted and compressed concurrently, and then embedded into a carrier image. However, existing schemes have some performance limitations in the quality of the reconstructed and cipher images and the efficiency. To address above issues, this work proposes a new visually secure image encryption scheme. First, we devise an adaptive-thresholding sparsification to greatly improve the quality of the reconstructed image. Second, we design a new parallel CS technique to tremendously improve the processing efficiency. Further, a matrix encoding strategy is finally employed to significantly reduce the number of changed bits in embedding process. Simulations and comparisons show that our proposed scheme has a high security level. Meanwhile, it is also more efficient, and achieves higher quality of the reconstructed and cipher images than some newly developed schemes.

Keywords: Image security; image compression; separable wavelet transform; parallel compressive sensing.

1. Introduction

As a typical kind of multimedia data, the digital images are widely used to deliver information every moment. Because a digital image can contain many potential information, it may cause a serious information security incident when some secret images, e.g. military images, are acquired by unauthorized accesses. Therefore, it is very important to protect the contents of secret images.

Among all the technologies of protecting images, the encryption is the most straightforward and effective one [1, 2, 3, 4]. Many image encryption algorithms have been developed and these algorithms can be divided into three kinds [5, 6, 7, 8]. The first kind only encrypts a plain image to an unrecognizable image with the same size using different techniques including chaos theory [9, 10, 11, 12], DNA coding [13, 14], quantum transformation [15, 16], cellular automata [17, 18], domain transformation [19, 20] and etc. However, this kind of algorithms has some obvious weaknesses. First, the encryption structures of some algorithms have security defects and thus the encrypted results have a high risk to be successfully broken [21]. Besides, the cipher images in these encryption algorithms usually have

the same sizes with the plain images, and this leads to low encryption efficiency. Finally, the generated random-like cipher images are more likely to attract the attentions of attackers.

The second kind of encryption algorithms encrypts a plain image to be a unrecognizable image with reduced size. To concurrently perform image compression and encryption, many researchers have introduced the compressive sensing (CS) technology into the image encryption [22, 23]. For example, the authors in [24] first compressed a plain image using the CS, and then encrypted the compressed image by scrambling and diffusing image pixels. To improve the security level, the authors in [25] further proposed a parallel CS (PCS) technique to resist chosen plain attack. These encryption algorithms can concurrently compress and encrypt a plain image and thus has wide applications in many scenes. However, they also transform plain images to unrecognizable cipher images and this still cannot reduce the attentions of attackers.

To overcome the weaknesses of the previous two kinds of encryption algorithms, the third kind of encryption algorithms aims to encrypt a plain image to a cipher image with visual security [26]. These encryption algorithms usually include two stages: encryption stage and embedding stage. The encryption stage encrypts a plain image to a secret image, while the embedding stage embeds the secret image into a carrier image to generate the final visually meaningful cipher image. For example, the scheme in [27] first encrypts a plain image to an unrecognizable secret image, and then embeds the secret image into a carrier image by replacing the partial of the carrier image. To reduce the embedding size, Chai *et al*. proposed an encryption scheme using CS and discrete wavelet transform (DWT) [28]. In this encryption scheme, a plain image is first compressed using CS, and then encrypted to be a secret image, and finally embedded into a carrier image that has the same size with the plain image. To improve the security level and quality of the reconstructed image, the authors in [29] modified the encryption structure in [28] by introducing a new CS counter mode and integer wavelet transformation. Although these existing visually secure image encryption schemes can achieve a relatively high performance, they still have many performance limitations in the quality of the reconstructed and cipher images, and the processing efficiency.

To overcome the performance limitations of existing image encryption algorithms, this paper proposes a new visually secure image encryption scheme using adaptive-thresholding sparsification and PCS. First, a plain image is decomposed by separable wavelet transform (SWT) [30] and scrambled by the 2D cat map [31]. Secondly, the scrambled image is sampled using PCS with adaptive threshold for each column, and the measurement matrices for each column are generated by a chaotic system. Finally, after quantifying and diffusing, the secret image is embedded into a carrier image using matrix encoding technique. The contributions and novelty of this paper are summarized as follows:

- To improve the sparsification performance, we propose the adaptive-thresholding sparsification strategy, which utilizes SWT and column-based adaptive thresholding (CBAT).
- To improve the efficiency, we generate a random-order Bernoulli random matrix for each column in the data sampling and develop a new PCS technique.

- To reduce the data loss of carrier image, we introduce the matrix encoding technique to embed the secret image to the carrier image.
- Simulation and comparison results demonstrate that our proposed scheme is more efficient, and can achieve higher quality of the reconstructed and cipher images than some newly developed schemes.

The rest of this paper is organized as follows. Section 2 introduces some related works and preliminaries. Section 3 presents the proposed visually secure encryption scheme. Section 4 simulates the proposed scheme and analyzes its performance. Section 5 evaluates the security of the proposed scheme and compares it with some recently developed schemes. Section 6 gives a conclusion of this paper.

2. Related Works and Contributions

This section first introduces the CS theory, and then analyzes some representative visually secure encryption schemes using CS, and finally presents some techniques that are used to design new encryption scheme in Section 3.

2.1. CS Theory

The CS theory specifies that when sampling a sparse signal below the Nyquist rate, the original signal can be well recovered from the sampled data [32, 33]. It tells that a sparse signal can be represented by some projections that are far smaller than the original signal. Suppose a sparse signal $\mathbf{x}$ is of size $N \times 1$ and a measurement matrix $\mathbf{\Phi}$ is of size $M \times N$ ($M \ll N$), the sample process can be defined as

$$\mathbf{y} = \mathbf{\Phi} \cdot \mathbf{x}, \tag{1}$$

where $\mathbf{y}$ is the measurement vector with size $M \times 1$. The compression ratio (CR) is defined as $CR = M/N$. Usually, a natural signal is not sparse. To deal with a natural signal using CS theory, we first transform the signal into a frequency domain to get its sparse representation and then perform the CS to the obtained sparse signal. Thus, for a natural signal $\mathbf{s}$, the sample process can be defined as

$$\mathbf{y} = \mathbf{\Phi} \cdot \mathbf{\Psi} \cdot \mathbf{s} = \mathbf{\Phi} \cdot \mathbf{x}, \tag{2}$$

where $\mathbf{\Psi}$ is a sparse transformation matrix and $\mathbf{x}$ is the sparse representation of signal $\mathbf{s}$.

The sparse signal $\mathbf{x}$ is called K-sparse when it has K non-zero entries. To completely recover the original signal, the variables K, M and N should meet $M = O(K \log_2(N/K))$ [32, 33] and the measurement matrix $\mathbf{\Phi}$ should satisfy the restricted isometry property (RIP) [34]. When recovering the original signal, the estimation of $\mathbf{x}$, denoted as $\hat{\mathbf{x}}$, can be calculated by solving the following optimization problem

$$\min \|\hat{\mathbf{x}}\|_0 \quad \text{s. t.} \quad \mathbf{y} = \mathbf{\Phi} \cdot \hat{\mathbf{x}}, \tag{3}$$

where $\|\cdot\|_0$ denotes the l_0-norm. However, l_0 optimization problem is NP-hard and thus l_0-norm is usually replaced by l_1-norm. Recently, researchers have proposed many effective reconstruction methods, including the orthogonal matching pursuit (OMP) [35] and smoothed l_0 norm (SL0) [36].

2.2. Visually Secure Image Encryption Using CS

Many visually secure image encryption schemes have been developed using CS![28, 37, 29]. These algorithms first encrypt a plain image to be a secret image with reduced size, and then embed the secret image into a carrier image to generate a cipher image. Fig. 1 presents a general framework of these algorithms, and the whole encryption scheme is divided into three stages: sparsification, compressive sampling and embedding.

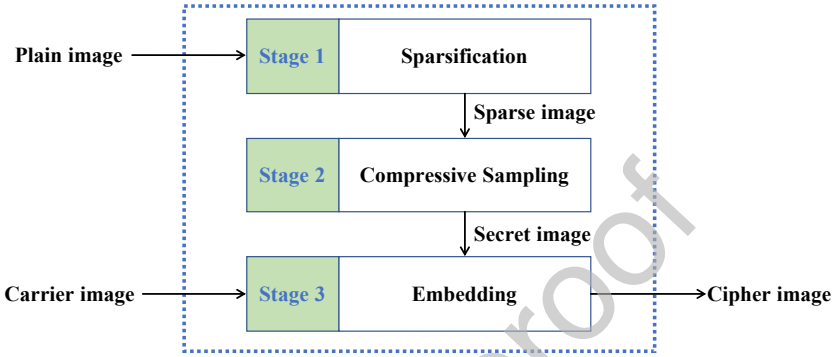


Figure 1: A general framework of the visually secure image encryption schemes using CS.

The sparsification is to transform a natural image to be a sparse signal before sampled. One of the most effective method is to transform the image from spatial domain to frequency domain using some techniques such as the wavelet transform and Fourier transform. Since a digital image has much data redundancy, its frequency spectrum has many elements that are close to 0. After setting these elements to 0 via quantifying with a threshold, a sparse signal can be generated. The compressive sampling compressively samples the sparse image to reduce the image size. According to the CS theory in Eq. (1), a measurement matrix is required when compressively sampling a sparse image. To solve the low security level of the secret image caused by a fixed measurement matrix for the whole sparse image [25], researchers usually use changeable measurement matrices generated by a chaotic system. After compressive sampling, a secret image is generated and then embedded into a carrier image to obtain visually meaningful cipher image. Usually, a pre-processing is performed to the carrier image to further improve the embedding space or reduce the data loss.

2.3. Our Contributions for New Scheme

We devise some new techniques to improve the quality of the reconstructed and cipher images and efficiency.

2.3.1. Adaptive-thresholding sparsification

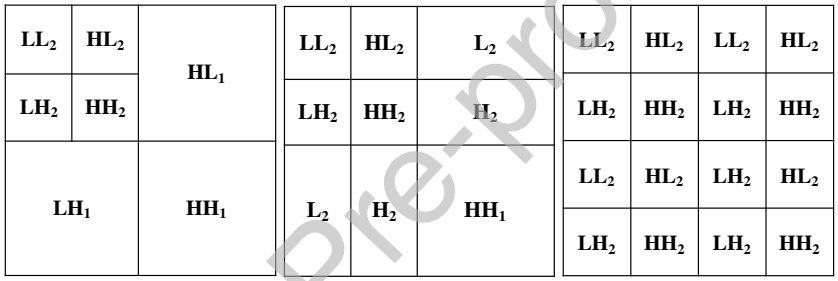
To improve the sparsification performance, an adaptive-thresholding sparsification strategy is introduced and it includes three operations: SWT, matrix confusion, and CBAT. The SWT is used to decompose the plain image to get its sparse representation in wavelet domain. First, the orthogonal wavelet transform matrix is calculated as

$$\mathbf{\Psi} = \mathbf{L}_n \mathbf{L}_{n-1} \dots \mathbf{L}_2 \mathbf{L}_1, \tag{4}$$

where the $\mathbf{L}_i$ ($i = 1, 2, \ldots, n$) of size $N \times N$ is calculated as

$$\mathbf{L}_i = \begin{bmatrix} \begin{bmatrix} H_{(N/2^i)\times(N/2^{i-1})} \\ G_{(N/2^i)\times(N/2^{i-1})} \end{bmatrix} & 0 \\ 0 & \mathbf{I} \end{bmatrix}, \tag{5}$$

where H and G are the low and high pass filters of the wavelet base, respectively. The maximum value of i is $\lfloor \log_2(N) \rfloor$ ($\lfloor \cdot \rfloor$ is to obtain the biggest integer that is not bigger than $\cdot$) and $\mathbf{I}$ is an identity matrix. Besides, the DWT and wavelet packet transform (WPT) are also widely used to decompose an image. However, the DWT has high processing efficiency but low sparsification performance, while the WPT has high sparsification performance but low processing efficiency. The SWT can well balance the trade-off between the performance and efficiency. Fig. 2 shows the 2-layer decomposition of the DWT, SWT and WPT. It is obvious that the SWT can better separate the high-frequency part and low-frequency part of an image than the DWT. Thus, it can achieve better sparsification performance.



(a) DWT (b) SWT (c) WPT

Figure 2: The 2-layer decomposition of an image using DWT, SWT and WPT.

After image decomposition by SWT, the elements in the coefficient matrix are divided into the principal and secondary components. The matrix confusion randomly shuffles the elements in the coefficient matrix such that the principal components can be uniformly distributed in each column. Different matrix confusion methods can be used to confuse the coefficient matrix, and the selection of confusion methods cannot affect the reconstruction quality. In our algorithm, we use the 2D cat map to shuffle the coefficient matrix since it is easy to be implemented and can obtain good confusion performance.

Finally, the CBAT is proposed to adaptively set thresholds for different images. Previous works usually set a global threshold and the pixel values smaller than the threshold are set to zero. However, since different images have different features, setting a fixed global threshold to all the images may lead to unstable performance. To solve these problems, we propose an adaptive method to directly set the sparsity for every column of image rather than set a threshold. In this case, the sparsity is fixed for every column, which can lead to a stable result. For a compression rate CR and column length N, we empirically set the sparsity of a column as

$$K = \lfloor \theta \cdot \omega \cdot CR^{\omega} \cdot e^{-\theta \cdot CR^{\omega}} \cdot N \rfloor, \tag{6}$$

where θ and ω are determined by the reconstruction method. Then the threshold for a column is the K-th largest absolute value in the column. The elements whose absolute values are smaller than the threshold will be set to zero. Note that the sparsity calculated by Eq. (6) may not ensure a best reconstruction quality for every image. However, this method can offer a simple way to dynamically set the thresholds for different images and avoid the unstable performance caused by a global threshold.

2.3.2. Chaos-Based Measurement Matrices

According to the discussions in [34], the measurement matrix in the CS should satisfy the restricted isometry property. A Bernoulli random matrix has the high probability to satisfy the restricted isometry property [38] and the chaotic matrix generated by the Logistic map is the Bernoulli random matrix [39]. In this work, we use the Logistic map to generate Bernoulli random matrix. The Logistic map is defined as

$$x_{n+1} = F(x_n) = \mu * x_n * (1 - x_n), x_n \in [0, 1], \tag{7}$$

where the control parameter $\mu \in [3.57, 4]$.

To generate C measurement matrices with size $M \times N$, a chaotic matrix $\mathbf{W}$ with size $M \times 2N$ is firstly generated by iterating the Logistic map with the given initial state (x_0) and quantifying the elements of generated chaotic sequence into 1 or -1 by the equation

$$t_i = \Omega(x_i) = \begin{cases} -1, & x_i \le 0.5 \\ 1, & x_i > 0.5. \end{cases} \tag{8}$$

Then, each of the C measurement matrices can be generated by randomly selecting N columns from $\mathbf{W}$ according to the chaotic sequences. Algorithm 1 shows the generation of the C measurement matrices $\mathbf{\Phi}_1, \mathbf{\Phi}_2, \cdots, \mathbf{\Phi}_C$.

2.3.3. Matrix Encoding Embedding

To achieve a better embedding performance, the matrix encoding [40] is introduced to embed the secret image into carrier image. Using more bits, the matrix encoding can represent a number of information bits with acceptable data loss. It can be described using a triple (n, k, t), where n is the representing bit number, k ($k \le n$) is the bit number to be represented and t is the maximum changed bit number. Suppose that a codeword $b = \{b_1 b_2 \dots b_n\}$ is the bits that can be changed in a block, $x = \{x_1 x_2 \dots x_k\}$ contains the secret bits and $b' = \{b'_1 b'_2 \dots b'_n\}$ is the modified codeword having embedded secret bits. Then the encoding process can be described as

Step 1: A function f is defined as

$$f(b) = (b_1 \times 1) \oplus (b_2 \times 2) \oplus \cdots \oplus (b_n \times n) = \hat{x}, \tag{9}$$

where $\oplus$ means bitwise xor operation.

Step 2: Find the position where the bit needs to be changed by

$$s = f(b) \oplus x \tag{10}$$

Algorithm 1: Generation of the C measurement matrices.

Input: Initial state (x_0), the number of measurement matrix C and matrix size $M \times N$.

Output: C measurement matrices $\mathbf{\Phi}_1, \mathbf{\Phi}_2, \cdots, \mathbf{\Phi}_C$.

Generate chaotic sequences $\mathbf{S} = \{x_1, x_2, \cdots, x_{2MN+2CN}\}$ using the Logistic map with (x_0);
Convert elements of $\mathbf{S}(1:2MN)$ into 1 or -1 by Eq. (8);
Initialize $\mathbf{W} = \mathbf{S}(1:2MN)$ and rearrange $\mathbf{W}$ as size $M \times 2N$ in row order;
Initialize $\mathbf{P} = \mathbf{S}(2MN+1:2MN+2CN)$ and rearrange $\mathbf{P}$ as size $C \times 2N$;
Initialize $\mathbf{\Phi}_1, \mathbf{\Phi}_2, \cdots, \mathbf{\Phi}_C \in \mathbb{R}^{M \times N}$;
for $i = 1 : C$ **do**
    $[\mathbf{P}', \mathbf{I}] = \text{SortI}(\mathbf{P}(i, :))$ {Sort the i-th row and $\mathbf{I}$ is the index vector.};
    for $j = 1 : N$ **do**
        $\mathbf{\Phi}_i(:, j) = \mathbf{W}(:, \mathbf{I}(j))$;
    end
end

Step 3: The rule to change the codeword b is illustrated as

$$b' = \begin{cases} b, & s = 0 \\ \{b_1, b_2, \ldots, 1 - b_i \ldots, b_n\}, & s = i \end{cases} \tag{11}$$

Replace b by b'.

Step 4: Repeat *Step 1* to *Step 3* until $f(b) = x$.

In this paper, the matrix encoding with $(n = 3, k = 2, t = 1)$ is used to embed a secret image into a carrier image. This means that we embed two bits of the secret image into the least three significant bits of a pixel in the carrier image, since the least significant bits contain less information. For example, suppose the two bits to be embedded is $x = \{10\}$ and a pixel in the carrier image is $\{10010110\}$. Then the least three significant bits are $b = \{110\}$. Following Eq. (9), we first calculate that $f(b) = (1 \times 1) \oplus (1 \times 2) \oplus (0 \times 3) = \{11\}$. Since $f(b) \neq x$, we calculate the position needed to be changed is $s = f(b) \oplus x = \{11\} \oplus \{10\} = 1$. Thus, we change the first bit of b to obtain $b' = \{010\}$. Then $f(b') = (0 \times 1) \oplus (1 \times 2) \oplus (0 \times 3) = \{10\}$. Because $f(b') = x$, the embedding process is finished and the modified pixel of value $\{10010010\}$ is obtained by replacing b with b'. In the data extraction stage, the embedded two bits can be extracted by calculating $f(b') = \{10\}$. Besides, the 2D Logistic-adjusted-Sine map (2D-LASM) introduced in [41] is used to generate chaotic sequences, which will be used to control the embedding order to further improve the security. The 2D-LASM is defined as

$$\begin{cases} x_{i+1} = \sin(\pi\mu(y_i + 3)x_i(1 - x_i)), \\ y_{i+1} = \sin(\pi\mu(x_{i+1} + 3)y_i(1 - y_i), \end{cases} \tag{12}$$

where the parameter $\mu \in [0, 1]$ and (x_0, y_0) are initial values.

3. Visually Secure Image Encryption Scheme

In this section, we present a new visually secure image encryption scheme. In the encryption process, the plain image is first encrypted to be a secret image, which is then embedded into a carrier image to get the visually meaningful cipher image. Fig. 3 shows the structure of the proposed encryption scheme. It includes two stages: encryption and embedding. In the encryption stage, the SWT is used to decompose the plain image, the 2D cat map is used to randomly shuffle the pixels, and the CBAT is to generate sparse image, ensuring that each column has the same number of zero. The secret key generates the initial states, which are employed by the chaotic maps to generate the measurements matrices, and the parameters in diffusion and matrix encoding. The PCS sampling is performed to the sparse image in the column-wise manner. After performing the quantification and diffusion to the sampling result, a secret image can be obtained. In the embedding stage, the matrix encoding technique embeds the secret image into a carrier image under the control parameters generated by the 2D-LASM. In the decryption process, the secret image is first extracted from the cipher image, and then decrypted to obtain the plain image. The decryption scheme is the combination of the inverse operation of the encryption scheme and its structure is shown in Fig. 4.

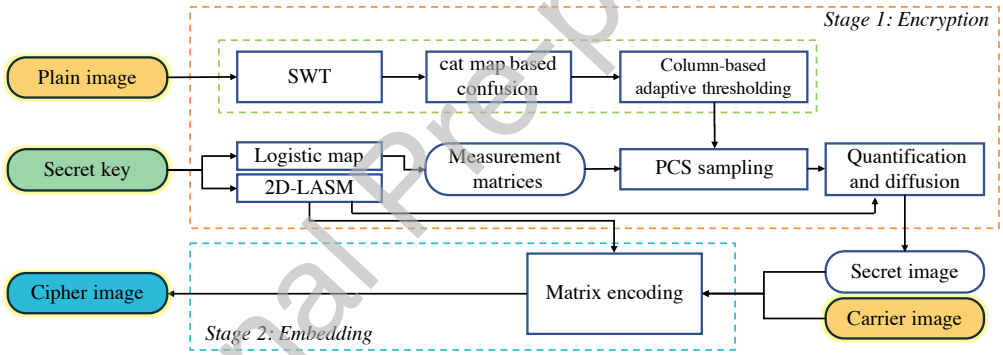


Figure 3: The structure of the proposed visually secure image encryption scheme.

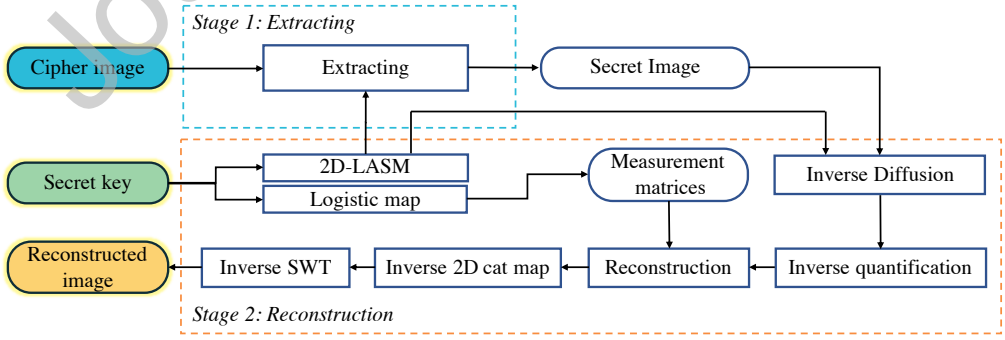


Figure 4: The structure of decryption scheme.

3.1. Secret Key

The secret key $\mathbf{K}$ is composed of 256 bits and it is used to generate the initial states of the Logistic map and 2D-LASM. The secret key should be transmitted over a secure channel or some public key encryption algorithms such as the well-known RSA. Firstly, a hash function SHA-256 is performed to the secret key to enhance the security level, and the hashed result contains six parts, namely $\mathbf{K}' = \{x_0, y_0, \mu, \gamma_1, \gamma_2, \gamma_3\}$, where (x_0, y_0, μ) are the original initial states, $(\gamma_1, \gamma_2, \gamma_3)$ are interference parameters. Algorithm 2 details the generation procedures of the initial states for the chaotic maps. Three groups of initial states $(x_0^{(1)})$, $(x_0^{(2)}, y_0^{(2)}, \mu^{(2)})$ and $(x_0^{(3)}, y_0^{(3)}, \mu^{(3)})$ can be obtained, and they are used in generating the measurement matrices, and the parameters in diffusion and matrix encoding, respectively.

Algorithm 2: The generation of the initial states for chaotic maps.

Input: Secret key $\mathbf{K}$ with length of 256 bits.

Output: Initial states $(x_0^{(1)})$, $(x_0^{(2)}, y_0^{(2)}, \mu^{(2)})$ and $(x_0^{(3)}, y_0^{(3)}, \mu^{(3)})$.

$\mathbf{K}' = \mathrm{SHA256}(\mathbf{K})$;

$x_0 = (\sum_{i=1}^{64} \mathbf{K}'[i] \times 2^{i-1})/2^{64}$;

$y_0 = (\sum_{i=65}^{128} \mathbf{K}'[i] \times 2^{i-65})/2^{64}$;

$\mu = (\sum_{i=129}^{192} \mathbf{K}'[i] \times 2^{i-129})/2^{64}$;

$\gamma_1 = (\sum_{i=193}^{213} \mathbf{K}'[i] \times 2^{i-193})/2^{21}$;

$\gamma_2 = (\sum_{i=214}^{234} \mathbf{K}'[i] \times 2^{i-214})/2^{21}$;

$\gamma_3 = (\sum_{i=235}^{256} \mathbf{K}'[i] \times 2^{i-235})/2^{22}$;

$x_0^{(1)} = ((x_0 + x_0 \times 2^5 \times \gamma_1) \mod 1) + 10^{-5}$;

for $i = 2 : 3$ **do**

  $x_0^{(i)} = ((x_0 + x_0 \times 2^{i\times 5} \times \gamma_i) \mod 1) + 10^{-5}$;

  $y_0^{(i)} = ((y_0 + y_0 \times 2^{i\times 5} \times \gamma_i) \mod 1) + 10^{-5}$;

  $\mu^{(i)} = ((\mu + \mu \times 2^{i\times 5} \times \gamma_i) \mod 0.4) + 0.5$;

end

3.2. Encryption and Reconstruction

Here, we describe the encryption process in the forward operation, and the related reconstruction process in the backward operation. Suppose the plain image $\mathbf{P}$ to be encrypted is of size $N \times N$. The detailed steps are described as follows:

Step 1: Apply the SWT introduced in Section 2.3.1 on $\mathbf{P}$ and the generated coefficient matrix $\mathbf{P1}$ with size $N \times N$ is calculated as

$$\mathbf{P1} = \mathbf{\Psi} \times \mathbf{P} \times \mathbf{\Psi}^T, \tag{13}$$

where $\mathbf{\Psi}$ is the orthogonal wavelet matrix computed by Eq. (4) with layer $\lfloor \log(N) \rfloor$. The inverse operation to recover the plain image $\mathbf{P}$ is

$$\mathbf{P} = \mathbf{\Psi}^T \times \mathbf{P1} \times \mathbf{\Psi}. \tag{14}$$

Step 2: The 2D cat map randomly shuffles the pixel positions of the image **P1**, and it is defined as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \mod N, \tag{15}$$

where (x, y) is a pixel position in the original image and (x', y') is a pixel position in the shuffled image, a and b are two parameters. Iterate the image **P1** c times using the 2D cat map to obtain a totally shuffled image **P2**. Note that the a, b and c are three parameters that can affect the reconstruction quality. The image **P1** can be recovered from **P2** by iterating the inverse 2D cat map c times using the same parameters, and the inverse 2D cat map is define as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ab+1 & -a \\ -b & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \mod N. \tag{16}$$

Step 3: Predefine the compression ratio of the plain image as CR. Apply the CBAT presented in Section 2.3.1 to **P2** to obtain the sparse image. Generate N measurement matrices, $\mathbf{\Phi}_1, \mathbf{\Phi}_2, \cdots, \mathbf{\Phi}_N$, using the Algorithm 1, where $M = N \times CR$ and the initial state for the Logistic map is $x_0^{(1)}$.

Step 4: The i-th measurement matrix $\mathbf{\Phi}_i$ is used to sample the i-th column of **P2** using the PCS. Then the compressed image **P3** with size $M \times N$ is generated. The image **P2** can be reconstructed from **P3** via different CS reconstruction methods.

Step 5: Quantify the pixel values of **P3** to be integer of range $[0, 255]$ to generate the quantified matrix **P4**. The quantification process is calculated as

$$\mathbf{P4} = \left\langle \frac{\mathbf{P3} - P_{\min}}{P_{\max} - P_{\min}} \times 255 \right\rangle, \tag{17}$$

where $P_{\min}$ and $P_{\max}$ are the minimum and maximum values of **P3**, respectively, and $\langle \cdot \rangle$ is to get the nearest integer. After the quantification, an image **P4** with integer pixel value is obtained. The inverse operation of quantification is defined as

$$\mathbf{P3} = \frac{\mathbf{P4} \times (P_{\max} - P_{\min})}{255} + P_{\min}. \tag{18}$$

Step 6: A diffusion operation is developed to randomly change the pixel value and spread the little change to the whole image. Specifically, a chaotic sequence $X = \{x_i\}_{i=1}^{MN}$ is generated by the 2D-LASM with initial state $(x_0^{(2)}, y_0^{(2)}, \mu^{(2)})$. Then convert the sequence $\mathbf{X}$ into integers $\mathbf{V} = \{v_1, v_2, \cdots, v_{MN}\}$ using the following operation

$$v_i = \langle x_i \times 2^{30} \rangle \mod 256. \tag{19}$$

Let $P4_i$ and s_i donate the ith element of **P4** and the ith element of the secret image $\mathbf{S}$. Then s_i can be calculated by

$$s_i = \begin{cases} P4_i \oplus v_i & i = 1 \\ P4_i \oplus v_i \oplus s_{i-1} & i > 1 \end{cases}. \tag{20}$$

Reshape the $\mathbf{S}$ with size $M \times N$ to obtain the secret image. The inverse operation is

$$P4_i = \begin{cases} s_i \oplus v_i & i = 1 \\ s_i \oplus v_i \oplus s_{i-1} & i > 1 \end{cases}. \tag{21}$$

3.3. Embedding and Extracting

After the plain image is encrypted to a secret image, the secret image is then embedded into a carrier image to further enhance the security level. Then a visually secure cipher image can be obtained by embedding the secret image $\mathbf{S}$ into the carrier image $\mathbf{Q}$. Suppose the size of the carrier image is $M_2 \times N_2$. To completely recover the secret image, the sizes of the carrier image and the secret image should satisfy that

$$M_2 \times N_2 \geq M \times N \times 4. \tag{22}$$

In the embedding, two chaotic sequences $\mathbf{X} = \{x_1, x_2, \cdots, x_{M_2N_2}\}$ and $\mathbf{Y} = \{y_1, y_2, \cdots, y_{M_2N_2}\}$ are generated using the 2D-LASM with the initial state $(x_0^{(3)}, y_0^{(3)}, \mu^{(3)})$. Select the last $4MN$ elements of $\mathbf{Y}$ to form another sequence $\mathbf{Y}' = \{y'_1, y'_2, \cdots, y'_{4MN}\}$. Sort the sequence $\mathbf{X}$ and $\mathbf{Y}'$ in increasing order to generate two index vectors $\mathbf{I}_x$ and $\mathbf{I}_y$. Then rearrange the secret image $\mathbf{S}$ as a vector $\{s_1, s_2, \cdots, s_{MN}\}$. By decomposing each pixel into 8 bits, a binary matrix with size $MN \times 8$ can be generated, and it is rearranged as $\mathbf{S}'$ with size $4MN \times 2$. Finally, the binary matrix $\mathbf{S}'$ can be embedded into the carrier image $\mathbf{Q}$ using matrix encoding under the control of $\mathbf{I}_x$, and $\mathbf{I}_y$. Algorithm 3 shows the pseudo-code of embedding the secret image $\mathbf{S}$ into the carrier image $\mathbf{Q}$ using the matrix encoding technique.

Algorithm 3: The procedure of embedding a secret image into a carrier image.

Input: The secret image $\mathbf{S}$ with size $M \times N$, the carrier image $\mathbf{Q}$ with size $M_2 \times N_2$, the index vectors $\mathbf{I_x}$ and $\mathbf{I_y}$.

Output: The visually secure cipher image $\mathbf{C}$.

Decompose each pixel of $\mathbf{S}$ into 8 bits, and rearrange its size to obtain a binary matrix $\mathbf{S}'$ with size $4MN \times 2$;
Decompose each pixel of $\mathbf{Q}$ into 8 bits, and rearrange its size to obtain a binary matrix $\mathbf{Q}'$ with size $M_2N_2 \times 8$;
for $i = 1 : 4MN$ **do**
    $x = \mathbf{I_x}(i)$ and $y = \mathbf{I_y}(i)$;
    $a = [\mathbf{Q}'(x, 6) \cdot 1] \oplus [\mathbf{Q}'(x, 7) \cdot 2] \oplus [\mathbf{Q}'(x, 8) \cdot 3]$;
    $b = a \oplus [\mathbf{S}'(y, 1) \cdot 2 + \mathbf{S}'(y, 2)]$ $(b \in [0, 3])$;
    if $b! = 0$ then change $\mathbf{Q}'(x, b + 5)$
end
Convert each row of $\mathbf{Q}'$ to be a decimal integer;
Rearrange $\mathbf{Q}'$ to obtain the cipher image $\mathbf{C}$ with size $M_2 \times N_2$.

Using the inverse operation of matrix encoding, the secret image $\mathbf{S}$ can be completely extracted from the cipher image $\mathbf{C}$ using the same index matrices $\mathbf{I}_x$ and $\mathbf{I}_y$.

3.4. Discussion

Because many effective techniques are introduced in our proposed scheme, the scheme is able to protect a plain image with a high security level and achieves many advantages.

First, the compression ratio and reconstruction quality of the original image can be greatly enhanced, due to the adaptive-thresholding sparsification. The SWT can decompose the plain image with high sparsification performance and processing efficiency, the generated measurement matrices can well satisfy the RIP, and the CBAT can ensure that every column of an image has the same sparsity and can be reconstructed with acceptable data loss. Thus, one can exactly set the compression ratio to the original image and obtain a high-quality reconstructed result.

Second, the cipher image can achieve high quality and has a similar data loss for different carrier images using the matrix encoding. Suppose the carrier image $\mathbf{I}$ and cipher image $\mathbf{C}$ are with size $M_2 \times N_2$, and the size of the secret image to be embedded is $M_2 \times N_2/4$, which is maximum value of satisfying the requirements of matrix encoding in Eq. (22). According to the embedding process described in Algorithm 3, the matrix coding is directly applied on bit level and cannot change the number of bits. Thus, this operation cannot cause data overflow. Meanwhile, each pixel in $\mathbf{I}$ has the same 25% probabilities to change one of its last three bits, or keep no change. This indicates $|\mathbf{I}(i,j) - \mathbf{C}(i,j)|$ can be 0, 1, 2, and 4 with the same 25% probabilities. Then the difference between the carrier image and the cipher image, denoted by the Peak Signal Noise Ratio (PSNR) [42], can be obtained as

$$\begin{aligned} PSNR &= 10 \times \log_{10}\left(\frac{255^2 \times M_2 \times N_2}{\sum_{i=1}^{M_2}\sum_{j=1}^{N_2}(\mathbf{I}(i,j) - \mathbf{C}(i,j))^2}\right) \\ &= 10 \times \log_{10}\left(\frac{255^2 \times M_2 \times N_2}{\sum_{i=1}^{M_2}\sum_{j=1}^{N_2} 0.25 \times (0^2 + 1^2 + 2^2 + 4^2)}\right) \\ &\approx 40.9292. \end{aligned} \tag{23}$$

This theoretically demonstrates that the embedding process has greatly high performance and the cipher image has few data loss compared with the carrier image.

Third, the scheme can well balance the trade-off between the size of the carrier image and the compression ratio CR. From Eq. (22), one can obtain that when the size of the carrier image is fixed, the compression ratio has a maximum value. Thus, one is flexible to set the compression ratio and the size of the carrier image. Besides, because the embedding process is a completely reversible operation, the secret image can be completely extracted from the cipher image without data loss.

Finally, the proposed scheme can protect the plain image with a high security level. This is because the PCS sampling, diffusion and matrix encoding are all under the control of the chaotic sequences generated by chaotic maps. Thus, the cipher image has a high security level and can resist the commonly used security attacks.

4. Simulation Results and Analysis

This section simulates the proposed visually secure image encryption scheme and analyzes its performance. To show a relatively fair simulation result, ten classical and widely used images are tested in our experiments and these images include "Brain", "Finger", "Girl", "Bridge", "Barbara", "Peppers", "Lena", "Jet", "Airplane", and "Baboon". The former two images have the size of 256×256, while the latter eight images have the size of 512×512. The proposed scheme is implemented using Matlab R2020a in a macOS Catalina 10.15.6 Operation system.

4.1. Simulation Results

This subsection simulates the proposed scheme using different images as the plain images and carrier images. The secret key is randomly generated as $\mathbf{K}$ = 9768*A*057*A*8159*D*63*E*6996*CC*6*B*23*CB*8*C*6*D*056*BA*0152*D*3*F*884*F*9*B*4*B*5*F*6147*B*33*DB*, the filter used in SWT is the sym8, parameters in 2D cat map are set as $(a, b, c) = (2, 3, 13)$, the compression ratio CR is set as 0.25 and the reconstruction method in PCS is the SL0 with parameters $\theta = 0.3445$ and $\omega = 1.404$ for the CBAT.

Fig. 5 shows the simulation results and each column is an individual experiment. Because the compression ratio $CR = 0.25$, the secret images are the 25% large as the plain image and they are noise-like. After embedding the secret images into the carrier images, the generated cipher images have the same visual effects with the carrier images. This can well protect the secret images because the meaningful images can greatly reduce the attentions of the attackers. Because the embedding process is completely invertible and the reconstruction of PCS has high performance, the reconstructed images have high quality and visually same effects with the plain images.

To qualitatively assess the performance of our proposed scheme, we use the PSNR presented in Eq. (23) and mean structural similarity (MSSIM) [43] to measure the quality of cipher images and reconstructed images. The MSSIM is to qualitatively describe the structural similarity of two images. The MSSIM of two images $\mathbf{X}$ and $\mathbf{Y}$ can be calculated by

$$MSSIM(\mathbf{X}, \mathbf{Y}) = \frac{1}{M} \sum_{k=1}^{M} \mathrm{SSIM}(x, y). \tag{24}$$

The structural similarity (SSIM) is calculated by

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + (k_1 \times L)^2)(2\sigma_{xy} + (k_2 \times L)^2)}{(\mu_x^2 + \mu_y^2 + (k_1 \times L)^2)(\sigma_x^2 + \sigma_y^2 + (k_2 \times L)^2)}, \tag{25}$$

where M is the number of image block, k_1 and k_2 are two parameters, L is the grayscale level and $L = 255$ for 8-bit grayscale image, x and y are blocks of images $\mathbf{X}$ and $\mathbf{Y}$, respectively, μ_i and σ_i $(i = x, y)$ are the average and variance values of block i, respectively, σ_{xy} is the covariance of blocks x and y. According to the recommendation in [43], we set $k_1 = 0.01$, $k_2 = 0.03$ and $M = 64$.

Table 1 shows the PSNR and MSSIM test results. As can be seen, all the PSNR values between the cipher and carrier images are above 40.9 dB and all the MSSIM values are above 0.99. This demonstrates that the cipher images have high similarity with the carrier images. Besides, all the reconstructed images also achieve high PSNR

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

Figure 5: Encryption and decryption results. The five rows from top to bottom are the plain images, secret images, carrier images, cipher images and reconstructed images, respectively.

Table 1: The PSNR and MSSIM values between the cipher and carrier images, and between the reconstructed and plain images.

Plain	Carrier	Cipher images		Reconstructed images	
images	images	PSNR(dB)	MSSIM	PSNR(dB)	MSSIM
Brain	Finger	40.9328	0.9967	37.9061	0.9481
Girl	Bridge	40.9295	0.9974	37.4035	0.9683
Barbara	Peppers	40.9310	0.9917	30.3344	0.9381
Lena	Jet	40.9186	0.9945	35.3992	0.9658
Airplane	Baboon	40.9187	0.9968	34.6182	0.9579

and MSSIM values from the plain images, and this indicates that the reconstructed images have good quality. Thus, our proposed scheme not only can concurrently compress and encrypt a plain image, but also generate cipher image with high quality to ensure visual security. With these significant properties, our proposed scheme has the potential to satisfy the requirements of visual security, compression ratio and reconstruction quality in practical applications.

4.2. Reconstruction Quality Against Compression Ratio

The quality of the reconstructed image is highly related to the compression ratio. To investigate the relationship between the reconstruction quality and compression ratio CR, we simulate our scheme using two different construction methods in the inverse operation of the PCS, namely the OMP and SL0. The experiment results show that the best reconstruction quality can be achieved when the parameters of CBAT are set as $\theta = 0.2576$, $\omega = 1.265$ in OMP, and are set as $\theta = 0.3445$, $\omega = 1.404$ in SL0. The plain image are selected as the "Barbara", "Airplane", "Girl" and "Lena", and the compression ratio is set as $CR = \{0.1, 0.2, \cdots, 0.8\}$.

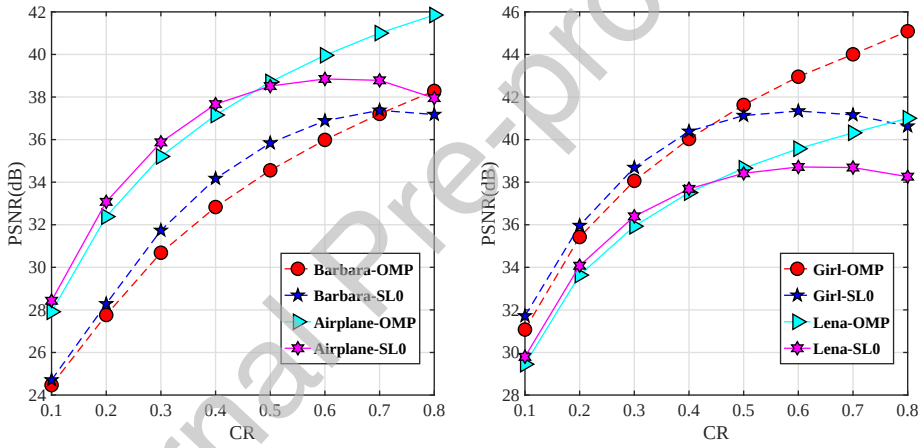


Figure 6: The PSNR values between the reconstructed and plain images with different CR and different reconstruction methods. (a) Images "Barbara" and "Airplane"; (b) Images "Girl" and "Lena".

Fig. 6 shows the PSNR values between the reconstructed and plain images with different CR and different reconstruction methods. From these figures, we can draw that when the CR is smaller than 0.5, the SL0 method can reconstruct the images with higher quality than the OMP method. In the contract, when the CR is bigger than 0.5, the OMP is more effective. Thus, the OMP reconstruction is more suitable for the light compression applications while the SL0 is more suitable for the heavy compression applications.

4.3. Carrier Image Against Reconstructed Image

Here, we test the performance of the proposed scheme for encrypting a plain image with different carrier images. Fig. 7 shows the visual quality of the cipher images and the reconstructed images using the images "Bridge", "Pep-

pers", "Jet" and "Baboon" as the carrier images, respectively. The plain image is the image "Lena". One can see that with different carrier images, the cipher images always have high visual quality and are similar to the corresponding carrier images. Besides, the reconstructed images can achieve a similar visual quality for different carrier images.

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

Figure 7: The quality of the reconstructed images against different different carrier images. (a)-(d) are four independent experiments, where the first, second and third rows are the carrier images, cipher images and reconstructed images, respectively.

To qualitatively test the effects of the carrier images, we calculate the PSNR and MSSIM values of the cipher images and the reconstructed images and Table 2 shows calculation results. As can be observed, all the PSNR and MSSIM values between the reconstructed and plain images are the same for different carrier images. This is because the embedding process is a completely reversible operation, the selections of the carrier images cannot affect the quality of the reconstructed image. These indicate that the proposed scheme is robust to the carrier image. Many similar schemes don't have this property and their performance highly depends on the selections of the carrier images. Our proposed scheme can overcome this weakness and one is flexible to select any digital image as the carrier image to achieve a high performance.

Table 2: The quality of the cipher and reconstructed images affected by different carrier images, where the plain image is the image "Lena".

Carrier	Cipher images		Reconstructed images	
image	PSNR(dB)	MSSIM	PSNR(dB)	MSSIM
Bridge	40.9293	0.9973	35.3992	0.9658
Peppers	40.9236	0.9918	35.3992	0.9658
Jet	40.9186	0.9945	35.3992	0.9658
Baboon	40.9278	0.9968	35.3992	0.9658

5. Security Analysis and Performance Comparison

This section analyzes the security level of the proposed encryption scheme and compares its performance with some newly developed schemes.

5.1. Key Security Analysis

An effective image encryption algorithm should have large enough key space to resist brute-force attack, and the ideal key space should be bigger than 2^{100} [44]. The secret key in our proposed scheme consists of 256 bits and its key space is 2^{256}, which is sufficient to satisfy the requirement of key space. Meanwhile, the secret key should be highly sensitive in both the encryption and decryption processes.

To test the sensitivity of the secret key, a secret key $\mathbf{K}_1$ is randomly generated to encrypt the plain image "Lena" and subsequently embed the secret image into a carrier image. By randomly changing one bit of the $\mathbf{K}_1$ in different positions, three new secret keys $\mathbf{K}_2, \mathbf{K}_3, \mathbf{K}_4$ are obtained as follows.

$$\mathbf{K}_1 = 7A09E5F4B5241E49B12CD5521E085A87F414A078E51C08D14535B487CBB3347A0,$$

$$\mathbf{K}_2 = 7A09E5F4B5241E49B12CD5521E085A87F414A078E51C08D14535B487CBB3347A\mathbf{1},$$

$$\mathbf{K}_3 = 7A09E5F4B5241E49B12CD5521E085A87F414A078E5\mathbf{3}C08D14535B487CBB3347A0,$$

$$\mathbf{K}_4 = 7A09E5F4B5241E4\mathbf{B}B12CD5521E085A87F414A078E51C08D14535B487CBB3347A0.$$

The four secret keys are separately used to decrypt a same cipher image, and Fig. 8 shows the decryption results. As can be seen, the decryption results using these incorrect secret keys are totally different with that using the correct secret key. Without secret key, one can't get any useful information about the plain image. Besides, the number of pixel change rate (NPCR) [45] is used to calculate the difference between the reconstructed images decrypted by the correct and incorrect keys. For two images $\mathbf{O}_1$ and $\mathbf{O}_2$ with the same size $M \times N$, their NPCR is defined as

$$NPCR = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\delta_{\mathbf{O}_1(i,j),\mathbf{O}_2(i,j)}}{MN} \times 100\%, \tag{26}$$

where $\delta_{\mathbf{O}_1(i,j),\mathbf{O}_2(i,j)}$ is 0 if $\mathbf{O}_1(i,j) = \mathbf{O}_2(i,j)$; otherwise it is 1. The NPCRs between the correct reconstructed image in Fig. (a) and incorrect reconstructed images in Figs. (b)-(d) are 0.9986, 0.9985 and 0.9987, respectively. This demonstrates that the proposed scheme has a high sensitivity on the secret key.

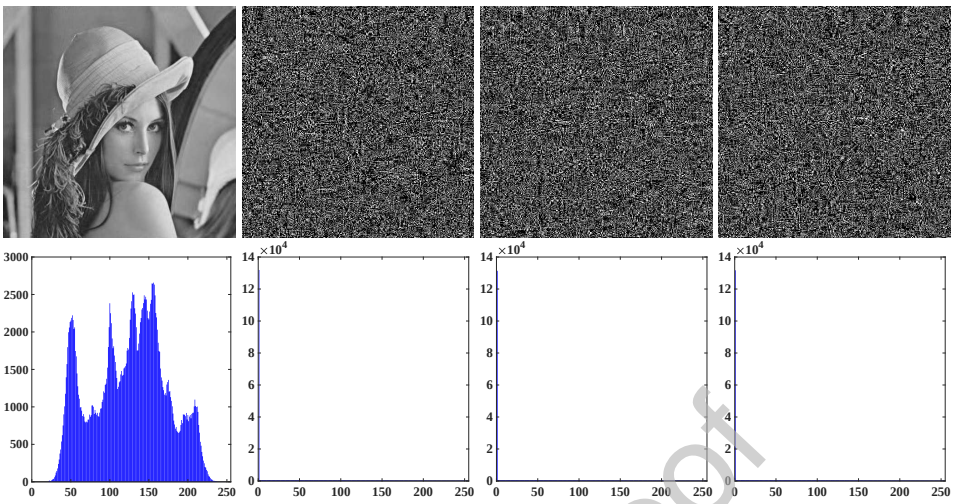


Figure 8: Key sensitivity analysis. Decryption results with (a) correct secret key $\mathbf{K}_1$ and (b)-(d) three incorrect secret keys $\mathbf{K}_2, \mathbf{K}_3, \mathbf{K}_4$.

In our proposed algorithm, the secret key is used to generate the measurement matrices and parameters for matrix coding, and different secret keys can result in totally different measurement matrices. However, since all the generated measurement matrices can satisfy the restricted isometry property, the reconstruction quality is almost the same for different measurement matrices. Thus, the reconstruction quality is stable for different secret keys. To test this property, we encrypt the image "Lean" using four different secret keys and the PSNRs between these generated four reconstructed image and the original image are 35.4398, 35.4617, 35.4602 and 35.3976, respectively. This experimentally indicates that the reconstruction quality is stable for different secret keys.

5.2. Histogram Analysis

The histogram of an image can directly reflect its statistical properties. For a natural image, its histogram usually has many patterns and thus one can obtain much useful information from it. From the information theory, one can get the least information when all the pixels distribute uniformly. Thus, an effective encryption algorithm should have the ability to generate secret images with uniform-distribution. Here, we use information entropy to measure the pixel distributions of the secret images, and the entropy H of a signal s is calculated as

$$H(s) = -\sum_{i=0}^{n} P(s_i) \log_2 P(s_i) \tag{27}$$

Where $P(s_i)$ is the probability of the ith possible value s_i. For an 8-bit grayscale image, n is 255 and the maximum entropy is 8 when the pixels are absolutely uniform-distributed.

We successively subject images "Peppers" and "Lena", "Baboon" and "Girl", "Jet" and "Airplane" to the proposed scheme. Fig. 9 shows the histograms of plain images and their secret images, and it is clear that the histograms of the

Table 3: The distances of histogram intersection between the carrier and cipher images.

Plain images	Peppers	Baboon	Jet
Carrier images	Lena	Girl	Airplane
Distance	0.9713	0.9613	0.9580

plain images have some patterns while the histograms of the secret images are uniform-distributed. One cannot get any useful information from the histograms of secret image. As for the information entropy, all secret images have the entropy that is very close to 8.

To achieve a higher security level, the cipher images are expected to have similar visual effects with the carrier images. Fig. 10 shows the histograms of the carrier images and their corresponding cipher images. One can see that the visual effects and histograms between the carrier image and its cipher image are quite similar. To qualitatively test the difference between the carrier and cipher images, the histogram intersection [46] is used to describe their similarity. For two histograms X and Y with N bins, their distance of histogram intersection is defined as

$$HI(X,Y) = \frac{\sum_{k=1}^{N} min(X_k, Y_k)}{\sum_{k=1}^{N} Y_k}. \tag{28}$$

A larger value means bigger similarity of the two histograms. Table 3 shows the distances of histogram intersection between the carrier images and cipher images. One can see that all the distances of histogram intersection are close to 1. This demonstrates the high similarity between the carrier images and its cipher images.

5.3. Adjacent Pixel Correlation

A natural image has strong correlation among the adjacent pixels. This correlation can benefit to the reconstruction of original image without secret key. Thus, a secret image is expected to have weak correlation among the adjacent pixels. Here, we evaluate the correlation of adjacent pixels using correlation coefficient. To calculate the correlation coefficient of an image, 3000 pixels are randomly selected in the image, and then the correlation coefficients between these pixels with their adjacent pixels in horizontal, vertical and diagonal direction are calculated.

Assume $X = \{x_i\}_{i=1}^{3000}$ and $Y = \{y_i\}_{i=1}^{3000}$ are two sequences of pixels and every pair (x_i, y_i) are adjacent pixels, the correlation coefficient of X, Y can be calculated as

$$CC_{XY} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}, \tag{29}$$

where σ_X donates the standard deviation of X and $Cov(X, Y)$ is the covariance of X and Y. Fig. 11 plots the adjacent pixel pairs of the plain, secret, carrier and cipher images, and Table 4 shows the numeral results. The used plain image is the image "Lena" and carrier image is the image "Jet". As can bee seen, the adjacent pixel pairs of the secret image are uniformly distributed in the whole phase plane, which is shown in Fig. 11(b), and their correlation coefficients are close to 0. This indicates that the adjacent pixels in the secret images have weak correlation. Besides, the cipher image has similar correlation coefficients with the carrier image, which implies the good similarity between the cipher and carrier image.

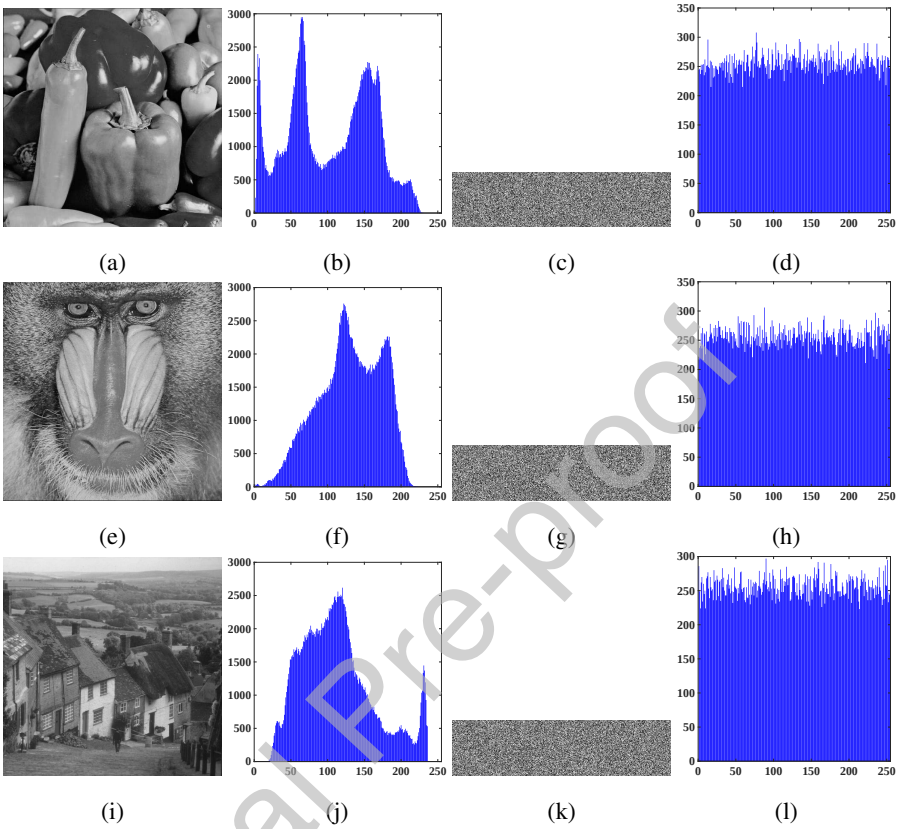


Figure 9: Histogram analysis about the plain images and their corresponding secret images. Pictures in these four columns represents plain images, histograms of plain images, secret images and histograms of secret images, respectively. The entropies of these three plain images are 7.5715, 7.3579 and 7.4778, while the entropies of their secret images are 7.9973, 7.9970 and 7.9976 respectively.

Table 4: Correlation coefficients along the horizontal, vertical and diagonal directions in the plain, secret, carrier and cipher images.

	Plain image	Secret image	Carrier image	Cipher image
Horizontal	0.9726	-0.0005	0.9729	0.9702
Vertical	0.9841	0.0173	0.9764	0.9716
Diagonal	0.9609	0.0007	0.9567	0.9493

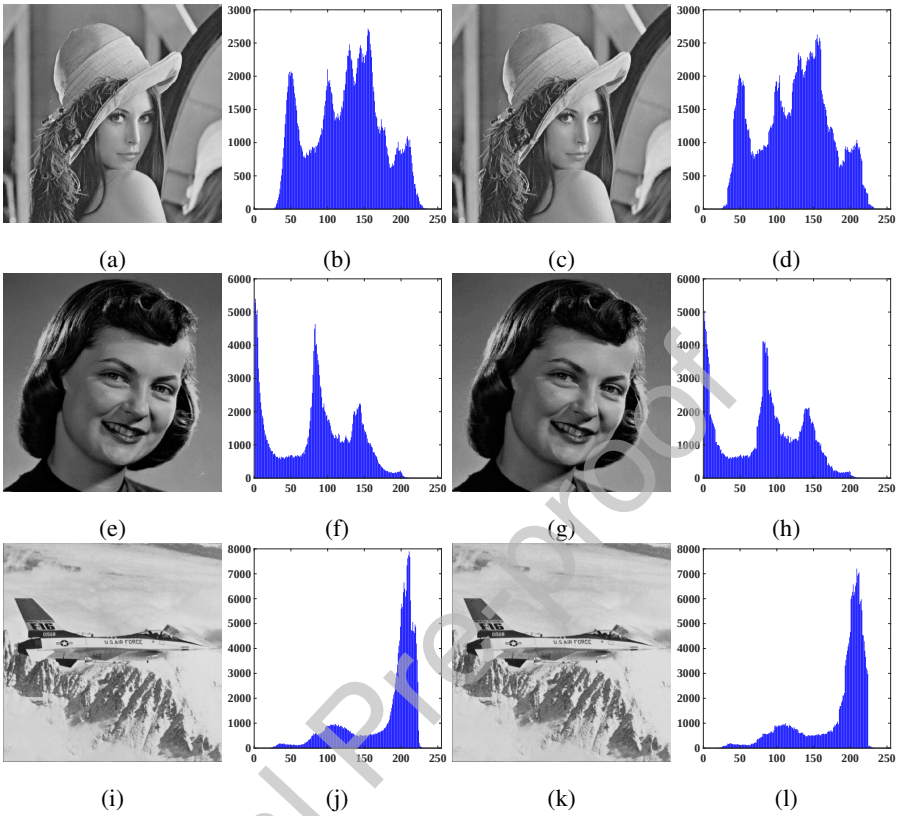


Figure 10: Histogram analysis about the carrier images and their corresponding cipher images. Pictures in these four columns represents carrier images, histograms of carrier images, cipher images and histograms of cipher images, respectively.

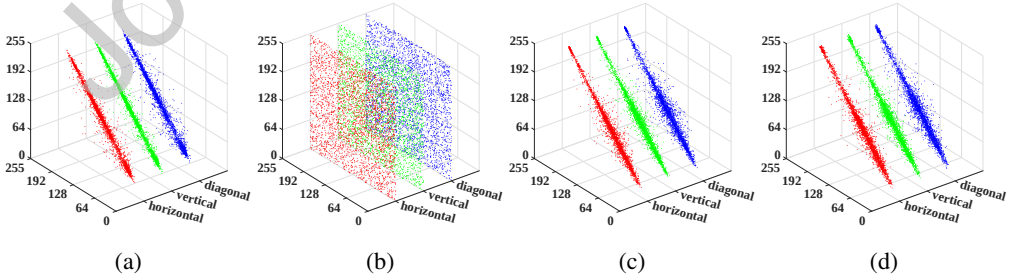


Figure 11: Adjacent pixel pairs along the horizontal, vertical and diagonal directions in the (a) plain image, (b) secret image, (c) carrier image and (d) cipher image.

5.4. Ability of Resisting Noise and Data Loss

Since almost all the transmission channels are noise channels, the cipher images should have the strong ability to resisting noise and data loss. This indicates that even a cipher image is blurred by the noise or has data loss, the decryption process can still recover the most information of the original image. Here, we test this ability of our proposed scheme.

Fig. 12 shows the experimental results with different percentages of data loss and salt and pepper noise. The image "Girl" is used as the plain image while the image "Jet" is used as the carrier image. Firstly, encrypt the plain image using the carrier image. Then, cause a cropping to the cipher image or blur the images. Finally, decrypt the cipher images with the correct key. One can see that the reconstructed images are still meaningful and readable. This indicates that even the data of the cipher image is changed in a certain level, the proposed scheme can still recover most information in the original image. Thus, the proposed scheme has a good robustness against noise pollution and cropping attack.

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

Figure 12: Simulation results for the ability of resisting data loss and noise. The first row shows the cipher images, while the second row shows the corresponding reconstructed images. (a) the original cipher image; (b) a 32×32 block cropping; (c) a 64×64 block cropping; (d) 0.1% salt and pepper noise; (e) 1% salt and pepper noise.

5.5. Efficiency Analysis

The fast increment of digital images requires high efficiency to the encryption process. Here, we theoretically analyze the time complexity of our proposed scheme and experimentally test its speed. Suppose the plain image is of size $N \times N$, the secret image is of size $M \times N$, and the label c_i in subsequently description represents a constant number. In the encryption stage, the total time complexity for the SWT, 2D cat map confusion and CBAT is $O(c_1 N^2)$. For PCS sampling, time mainly costs in the generation of measurement matrices, and the total time complexity is $O(c_2 N^2 \log(N))$. Since the quantification and diffusion are the linear operations, their time complexity is $O(c_3 MN)$.

In the embedding stage, because the generating and sorting of chaotic sequences require a proceeding time, the time complexity for the matrix encoding in the embedding stage is $O(c_4N^2\log(N^2))$. Among above complexities, $O(N^2\log(N^2))$ has the maximum magnitude, which will determine the actual running time of our proposed encryption scheme. Thus, the total computational complexity of encryption process is $O(cN^2\log(N^2))$. Meanwhile, the time complexity in the decryption process is highly determined by the reconstruction method.

To test the actual running time, the experiment is performed on a computer with Inter(R) Core(TM) i7-8700 @ 3.2GHz. Table 5 lists the encryption and decryption times for images with different sizes. It can be observed that the average encryption times is 0.0763s, 0.3328s and 1.9620s for images with different size 256×256, 512×512 and 1024×1024 respectively. This is roughly consistent with the theoretical results. On the other hand, the average decryption times are 0.3144s, 2.0590s and 52.7013s for images with different size 256×256, 512×512 and 1024×1024. With the increment of image size, the running time of decryption process grows fast. Thus, our proposed encryption scheme can achieve high encryption efficiency when the size of plain image is not large.

Table 5: The encryption and decryption times (second) of our proposed scheme for images with different sizes.

	Image size	Lena	Girl	Peppers	Baboon	Barbara	Airplane	Average
Encryption time	256×256	0.0747	0.0755	0.0776	0.0758	0.0760	0.0781	0.0763
	512×512	0.3472	0.3374	0.3283	0.3294	0.3283	0.3264	0.3328
	1024×1024	2.0319	1.9229	1.9723	1.9535	1.9510	1.9403	1.9620
Decryption time	256×256	0.2805	0.3675	0.3881	0.2937	0.2798	0.2766	0.3144
	512×512	2.0931	2.0913	2.0423	2.0465	2.0243	2.0566	2.0590
	1024×1024	53.6228	52.0911	49.9373	50.5455	49.9368	60.0746	52.7013

5.6. *Comparison with Latest Schemes*

To show the superiority of our proposed encryption scheme, we compare it with some other latest CS-based encryption schemes introduced in [28, 47, 48, 49, 50, 24, 29, 26]. The comparisons are performed from the aspects of the quality of the reconstructed image, the quality of the cipher image, and the efficiency. To provide a relatively fair comparison, the results of the competing schemes are all directly referenced from the original literatures.

The quality of the reconstructed image is the most important performance in an image encryption algorithm. Table 6 shows the comparisons of the PSNR values between the plain image and the reconstructed images by different encryption schemes. Since the results in [28, 49, 26] are shown in graphs, we estimate their PSNR values from the graphs. The N/A indicates that the value was not provided in the related literatures. As can been seen, under the same compression ratio CR, the proposed scheme can achieve the largest PSNR values for different images. This indicates that the reconstructed images by the proposed scheme have the highest quality. Note that our algorithm has a different reconstruction quality for a same image with different sizes. This is because our algorithm uses an adaptive way to

Table 6: Comparisons of the PSNR values between the plain and reconstructed images in different encryption schemes.

Images	*CR*	[28]	[47]	[48]	[49]	[50]	[24]	[29]	[26]	Proposed
Lena512	0.25	29.0562	N/A	N/A	28.5	31.4240	N/A	33.4204	28.5	**35.3992**
	0.5	35	N/A	N/A	33.5276	32.9660	34.5560	N/A	33	**38.4171**
Lena256	0.25	N/A	26.5600	26.0600	N/A	N/A	N/A	N/A	N/A	**29.3408**
	0.5	N/A	29.8300	29.8200	N/A	N/A	N/A	N/A	N/A	**35.4647**
Peppers	0.25	30.2	N/A	N/A	28	30.6809	N/A	N/A	N/A	**34.7013**
	0.5	34	N/A	N/A	33	31.9825	31.5132	N/A	N/A	**37.3215**

calculate the sparsity (shown in Eq. (6)) and the sparsity is related to the image column. The sparsity is different when two images have different sizes.

The cipher image in a visually secure image encryption scheme is expected to have high quality. We also compare the quality of the cipher images between our proposed scheme with the visually secure schemes introduced in [28, 37, 29]. Table 7 shows the comparisons of the PSNR and MSSIM values between the cipher and carrier images in different encryption schemes. One can see that the PSNR values of our proposed scheme for different images all approximate to 40.90 dB, which are much larger than the PSNR values of other encryption schemes. Besides, the MSSIM values of our proposed scheme are also larger than that of the other schemes. These demonstrate that our proposed scheme can generate a cipher image that is quite similar with the carrier image.

Table 7: Comparisons of the PSNR and MSSIM values between the cipher and carrier images in different encryption schemes.

Plain images	Carrier images	[28]		[37]		[29]		Proposed	
		PSNR(dB)	MSSIM	PSNR(dB)	MSSIM	PSNR(dB)	MSSIM	PSNR(dB)	MSSIM
Lena	Peppers	18.5136	0.6726	35.1347	*N/A*	32.3513	0.9257	**40.9115**	**0.9917**
Jet	Baboon	23.3967	0.6991	36.4906	*N/A*	37.8967	0.9833	**40.9286**	**0.9967**
Brain	Cameraman	24.8700	0.6488	35.3534	*N/A*	34.8967	0.9381	**40.9421**	**0.9602**
Girl	Airplane	28.2318	0.7021	36.2169	*N/A*	36.1125	0.9666	**40.9335**	**0.9895**
Barbara	Bridge	25.2321	0.7337	36.1070	*N/A*	35.5629	0.9783	**40.9233**	**0.9973**
Average		24.0488	0.6913	35.8692	*N/A*	35.2058	0.9584	**40.9278**	**0.9871**

Finally, we compare the encryption and decryption efficiencies between our proposed scheme and schemes introduced in [28, 47, 48, 24, 29]. The images "Finger" and "Baboon" with size 256×256 are used as the plain images. Table 8 shows the encryption and decryption times of different encryption schemes, respectively. The results show that the encryption and decryption speeds of our proposed scheme are much faster than the other five schemes, indicating the high efficiency of the proposed scheme.

Table 8: Comparisons results of encryption and decryption times (second) in different encryption schemes.

Schemes	Finger256		Baboon256		Average	
	Encryption	Decryption	Encryption	Decryption	Encryption	Decryption
[28]	0.1159	2.2295	0.1181	2.3325	0.1170	2.2760
[47]	0.3356	1.5216	0.3319	1.5342	0.3333	1.5279
[48]	0.4536	1.1374	0.4607	1.1413	0.4572	1.1393
[24]	N/A	N/A	0.4296	0.9544	0.4296	0.9544
[29]	0.1544	2.2489	0.1523	2.2870	0.1533	2.2679
Proposed	**0.0731**	**0.2860**	**0.0683**	**0.3041**	**0.0707**	**0.2950**

6. Conclusion

This paper proposed a visually secure image encryption scheme using adaptive-thresholding sparsification and PCS techniques. The scheme includes the encryption and embedding stages. The encryption stage first decomposes a plain image using SWT and scrambles the image using the 2D cat map, and then samples the scrambled image using PCS with a threshold for each column, and finally quantifies and diffuses the image to obtain a secret image. The embedding stage embeds the secret image into a carrier image using the matrix encoding. The adaptive-thresholding sparsification can greatly improve the quality of the reconstructed image by utilizing the SWT and CBAT. The PCS with random-order Bernoulli random matrices is adopted to enhance the processing efficiency. Besides, the matrix encoding technique can result in superior visual effect of the cipher image and doesn't affect the quality of the reconstructed image. Experiment results demonstrate the high security and strong robustness of our proposed scheme. Comparison results show that our proposed scheme can achieve better performance than some other latest schemes in the quality of the reconstructed and cipher images, and the processing efficiency. In future work, we will aim to further improve the efficiency and reconstruction performance by performing the sampling and reconstruction using deep learning model.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1003805, Natural Scientific Research Innovation Foundation in Harbin Institute of Technology under Grant HIT.NSRIF.2020077, and National Natural Science Foundation of China under Grants 61701137 and 62071142.

References

References

[1] X. Wang, L. Feng, H. Zhao, Fast image encryption algorithm based on parallel computing system, Information Sciences 486 (2019) 340–358.

[2] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, H. Huang, Cross-plane colour image encryption using a two-dimensional logistic tent modular map, Information Sciences 546 (2021) 1063–1083.

[3] S. Wang, C. Wang, C. Xu, An image encryption algorithm based on a hidden attractor chaos system and the knuth–durstenfeld algorithm, Optics and Lasers in Engineering 128 (2020) 105995.

[4] C. Xu, J. Sun, C. Wang, An image encryption algorithm based on random walk and hyperchaotic systems, International Journal of Bifurcation and Chaos 30 (4) (2020) 2050060.

[5] Z. Hua, Y. Zhou, C.-M. Pun, C. P. Chen, Image encryption using 2d logistic-sine chaotic map, in: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2014, pp. 3229–3234.

[6] R. Lan, J. He, S. Wang, T. Gu, X. Luo, Integrated chaotic systems for image encryption, Signal Processing 147 (2018) 133–145.

[7] N. Zhou, H. Jiang, L. Gong, X. Xie, Double-image compression and encryption algorithm based on co-sparse representation and random pixel exchanging, Optics and Lasers in Engineering 110 (2018) 72–79.

[8] C. Wang, H. Xia, L. Zhou, A memristive hyperchaotic multiscroll Jerk system with controllable scroll numbers, International Journal of Bifurcation and Chaos 27 (06) (2017) 1750091.

[9] Z. Hua, Y. Zhang, Y. Zhou, Two-dimensional modular chaotification system for improving chaos complexity, IEEE Transactions on Signal Processing 68 (2020) 1937–1949.

[10] R. Lan, J. He, S. Wang, Y. Liu, X. Luo, A parameter-selection-based chaotic system, IEEE Transactions on Circuits and Systems II: Express Briefs 66 (3) (2018) 492–496.

[11] A. Mansouri, X. Wang, A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme, Information Sciences 520 (2020) 46–62.

[12] Z. Hua, B. Zhou, Y. Zhou, Sine-transform-based chaotic system with FPGA implementation, IEEE Transactions on Industrial Electronics 65 (3) (2017) 2557–2566.

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

[14] R. Guesmi, M. A. B. Farah, A. Kachouri, M. Samet, A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2, Nonlinear Dynamics 83 (3) (2016) 1123–1136.

[15] L.-H. Gong, X.-T. He, S. Cheng, T.-X. Hua, N.-R. Zhou, Quantum image encryption algorithm based on quantum image XOR operations, International Journal of Theoretical Physics 55 (7) (2016) 3234–3250.

[16] N. Zhou, Y. Hu, L. Gong, G. Li, Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations, Quantum Information Processing 16 (6) (2017) 164.

[17] A. Y. Niyat, M. H. Moattar, M. N. Torshiz, Color image encryption based on hybrid hyper-chaotic system and cellular automata, Optics and Lasers in Engineering 90 (2017) 225–237.

[18] Y. Wang, Y. Zhao, Q. Zhou, Z. Lin, Image encryption using partitioned cellular automata, Neurocomputing 275 (2018) 1318–1332.

[19] S. Liansheng, X. Meiling, T. Ailing, Multiple-image encryption based on phase mask multiplexing in fractional Fourier transform domain, Optics Letters 38 (11) (2013) 1996–1998.

[20] Y. Luo, M. Du, J. Liu, A symmetrical image encryption scheme in wavelet and time domain, Communications in Nonlinear Science and Numerical Simulation 20 (2) (2015) 447 – 460.

[21] H. Wang, D. Xiao, X. Chen, H. Huang, Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map, Signal Processing 144 (2018) 444–452.

[22] Y. Zhang, P. Wang, H. Huang, Y. Zhu, D. Xiao, Y. Xiang, Privacy-assured fogcs: Chaotic compressive sensing for secure industrial big image data processing in fog computing, IEEE Transactions on Industrial Informatics (2020) 1–1.

[23] J. Wang, L. Y. Zhang, J. Chen, G. Hua, Y. Zhang, Y. Xiang, Compressed sensing based selective encryption with data hiding capability, IEEE Transactions on Industrial Informatics 15 (12) (2019) 6560–6571.

[24] Y. Luo, J. Lin, J. Liu, D. Wei, L. Cao, R. Zhou, Y. Cao, X. Ding, A robust image encryption algorithm based on Chua's circuit and compressive

sensing, Signal Processing 161 (2019) 227–247.

[25] R. Fay, C. Ruland, Compressive sensing encryption modes and their security, in: 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE, 2016, pp. 119–126.

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

[27] L. Bao, Y. Zhou, Image encryption: Generating visually meaningful encrypted images, Information Sciences 324 (2015) 197–207.

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

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

[30] H. Wang, J. Vieira, 2-D wavelet transforms in the form of matrices and application in compressed sensing, 2010 8th World Congress on Intelligent Control and Automation (2010) 35–39.

[31] G. Chen, Y. Mao, C. K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, Chaos, Solitons & Fractals 21 (3) (2004) 749–761.

[32] E. J. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Transactions on Information Theory 52 (2) (2006) 489–509.

[33] D. L. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (4) (2006) 1289–1306.

[34] E. J. Candes, T. Tao, Decoding by linear programming, IEEE Transactions on Information Theory 51 (12) (2005) 4203–4215.

[35] J. A. Tropp, A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Transactions on Information Theory 53 (12) (2007) 4655–4666.

[36] G. H. Mohimani, M. Babaie-Zadeh, C. Jutten, Fast sparse representation based on smoothed ℓ^0 norm, in: International Conference on Independent Component Analysis and Signal Separation, Springer, 2007, pp. 389–396.

[37] P. Ping, J. Fu, Y. Mao, F. Xu, J. Gao, Meaningful Encryption: Generating Visually Meaningful Encrypted Images by Compressive Sensing and Reversible Color Transformation, IEEE Access 7 (2019) 170168–170184.

[38] R. Lu, On the strong restricted isometry property of Bernoulli random matrices, Journal of Approximation Theory 245 (2019) 1–22.

[39] T. Sang, R. Wang, Y. Yan, Generating binary Bernoulli sequences based on a class of even-symmetric chaotic maps, IEEE Transactions on Communications 49 (4) (2001) 620–623.

[40] R. Crandall, Some notes on steganography, Posted on Steganography Mailing List (1998) 1–6.

[41] Z. Hua, Y. Zhou, Image encryption using 2D Logistic-adjusted-Sine map, Information Sciences 339 (2016) 237–253.

[42] J. Korhonen, J. You, Peak signal-to-noise ratio revisited: Is simple beautiful?, in: 2012 Fourth International Workshop on Quality of Multimedia Experience, IEEE, 2012, pp. 37–38.

[43] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.

[44] G. Alvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, International Journal of Bifurcation and Chaos 16 (08) (2006) 2129–2151.

[45] Y. Wu, J. P. Noonan, S. Agaian, et al., NPCR and UACI randomness tests for image encryption, Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT) 1 (2) (2011) 31–38.

[46] M. J. Swain, D. H. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.

[47] X. Chai, H. Wu, Z. Gan, Y. Zhang, Y. Chen, K. W. Nixon, An efficient visually meaningful image compression and encryption scheme based on compressive sensing and dynamic LSB embedding, Optics and Lasers in Engineering 124 (2020) 105837.

[48] X. Chai, X. Zheng, Z. Gan, D. Han, Y. Chen, An image encryption algorithm based on chaotic system and compressive sensing, Signal Processing 148 (2018) 124–144.

[49] J. Chen, Y. Zhang, L. Qi, C. Fu, L. Xu, Exploiting chaos-based compressed sensing and cryptographic algorithm for image encryption and

compression, Optics & Laser Technology 99 (2018) 238–248.

[50] Z. Gan, X. Chai, J. Zhang, Y. Zhang, Y. Chen, An effective image compressionencryption scheme based on compressive sensing (CS) and game of life (GOL), Neural Computing & Applications (2020) 1–29.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Zhongyun Hua: Conceptualization, Methodology, Supervision, Project administration, Writing - Original draft, Funding acquisition, Project administration
Kuiyuan Zhang: Term, Methodology, Software, Data curation, Validation, Investigation, Writing - Original draft, Visualization,
Yuanman Li: Writing - Review & Editing, Funding acquisition, Data curation, Methodology.
Yicong Zhou: Formal analysis, Resources, Writing - Review & Editing.