Research Article

Meaningful Image Encryption Based on Reversible Data Hiding in Compressive Sensing Domain

Ming Li ^{1,2} Haiju Fan ^{1,3} Hua Ren ¹ Dandan Lu ¹ Di Xiao ⁴ and Yang Li ^{2,5,6}

¹College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China

²School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

³China National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China

⁴College of Computer Science, Chongqing University, Chongqing 400044, China

⁵Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China

⁶Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, Beihang University, Beijing 100083, China

Correspondence should be addressed to Yang Li; liyang@buaa.edu.cn

Received 24 July 2017; Revised 10 October 2017; Accepted 16 January 2018; Published 14 February 2018

Academic Editor: Bruce M. Kapron

Copyright © 2018 Ming Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel method of meaningful image encryption is proposed in this paper. A secret image is encrypted into another meaningful image using the algorithm of reversible data hiding (RDH). High covertness can be ensured during the communication, and the possibility of being attacked of the secret image would be reduced to a very low level. The key innovation of the proposed method is that RDH is applied to compressive sensing (CS) domain, which brings a variety of benefits in terms of image sampling, communication and security. The secret image after preliminary encryption is embedded into the sparse representation coefficients of the host image with the help of the dictionary. The embedding rate could reach 2 bpp, which is significantly higher than those of other state-of-art schemes. In addition, the computational complexity of receiver is reduced. Simulations verify our proposal.

1. Introduction

Along with the development of Internet and information science, Internet can definitely provide great convenience to information transmission and processing, but as the same time, severe security problems therein have emerged gradually [1, 2]. Data hiding, as an important method to guarantee secure information transmission, has attracted more and more people's concern [3, 4]. It has solved the easily-being-attacked problems of Cryptosystem by hiding the existence of data. Most of the methods, however, arbitrarily alter the carrier of information, which result in an unrecoverable host image after the secret information has been extracted by the receiver, and the embedded information may also not be recovered exactly. This is vital in some distortion-unacceptable scenarios. Subsequently, the problem has been solved effectively by the reversible data hiding (RDH) [5–7]. RDH technique may hide the secret messages in digital images, and it has two merits: lossless restoration of host image and lossless extraction of embedded message.

Consequently, RDH has become one effective technique to achieve copyright protection for the host image and covert communication for the embedded message [8–10].

Image encryption is a well-known technique to protect the contents of the original images [11–13]. The encrypted result appears as a random image, whose histogram is flat and information entropy is close to 8. Therefore, a secure cryptographic system with large key space and effective encryption mechanism can resist unauthorized user to obtain the original image. But the noise-like image is easily detected and attacked by the hacker during the transmission. If the encryption result presents as a meaningful image, it would be secure visually, and the possibility of being attacked would be dropped down to a very low level. Recently, many researchers contribute to the visually secure encryption scheme [14, 15]. For example, Kanso and Ghebleh [14] first decomposed the host image into four sub-bands denoted by LL, HL, LH and HH, then embedded a scrambled image into the latter three sub-bands. Chai et al. [15] proposed to compress the shuffled plain image and embed it into the host image. In fact, the

method to encrypt a plain image into another meaningful image employs the method of data hiding; therefore they have used many common algorithms. Among them, RDH based on lossless compression [6, 16], RDH using difference expansion [17, 18] and RDH using histogram shifting [19, 20] are the most influential methods. It is noticed that, if the secret image is encrypted into a meaningful image, the aim of image encryption (another kind of RDH) is not to protect the host image any more, but to protect the embedded secret image. Therefore, meaningful image encryption can be used for covert communication of the secret image.

Compressive sensing (CS), which broke through the Shannon-Nyquist sampling theorem, brings a huge impact on the area of communication. Recently, CS and sparse representation are applied in RDH [15, 21–25]. Xiao et al. [22] embedded the secret data into the AC coefficient of host image and then used compressive sensing to generate the encrypted image. Yamaç et al. [23] proposed to embed the data into the CS measurements. Hua et al. [24] have proposed an informed sparse data hiding system, which embeds the message in the sparse domain and relaxes the sparsity condition. Meanwhile, many researchers tested that CS based RDH is secure to resist multiple attacks, because the measurement matrix or dictionary in CS based RDH is set to be secret. Rachlin and Baron [26] confirmed that it is not possible to reconstruct the signal using only the measurements without knowing the measurement matrix or dictionary. Orsdemir et al. [27] argued that brute attacks to obtain the matrix above were computationally infeasible. Therefore, processing the host image by CS effectively enhanced the security of RDH. In addition, it also broadened the applications of RDH. However, RDH in CS domain is still an attempt, and the computation complexities and the embedding capacities in the existing methods are still not satisfactory.

In this paper, enlightened by the idea of encrypting a secret image into another meaningful image which can significantly improve the covertness of the secret image communication, we propose a novel method of meaningful image encryption by combining RDH with CS. The host image is processed by CS, and the key problem of the proposed method is the achievement of RDH in CS domain. The contributions of this paper can be summarized as follows:

- (1) The covertness of the secret image to be encrypted can be ensured by RDH in a meaningful image.
- (2) A novel method of RDH in CS domain is proposed.
- (3) The embedding rate is significantly higher compared with the state-of-art schemes.
- (4) The complexity of computation of receiver is reduced.

The rest of this paper is organized as follows. Section 2 gives the related works about CS. Our proposed encryption scheme is described in Section 3. Section 4 verifies the proposed scheme via simulations. Comparisons between the proposed scheme and other works are discussed in Section 5. And the last section makes a conclusion of our work.

2. The Framework of CS

In this section, we discuss the basic CS theory and the exact recovery condition. Because the host images are usually

natural images, they can be compressed by CS theory. Let a host image and its sparse solution be $\mathbf{X} \in \mathbb{R}^{M \times 1}$ and $\mathbf{Y} \in \mathbb{R}^{N \times 1}$ respectively, then

$$\begin{aligned} \hat{\mathbf{Y}} &= \arg \min_{\mathbf{Y}} \|\mathbf{Y}\|_0 \\ \text{s.t. } \mathbf{X} &= \mathbf{D}\mathbf{Y}, \end{aligned} \quad (1)$$

where $\mathbf{D} \in \mathbb{R}^{M \times N}$ is a matrix with full row rank. If we want to reconstruct $\mathbf{Y}$ from $\mathbf{X}$, the mutual incoherence property (RIP) [28] must be obeyed, which is showed as follows

$$\mu \triangleq \max_{i \neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle|, \quad i, j = 0, 1, \dots, N-1, \quad (2)$$

where $\mathbf{d}_i$ and $\mathbf{d}_j$ are the i th and j th column vectors of $\mathbf{D}$ respectively. Tropp [28] has described the condition to reconstruct $\mathbf{Y}$ exactly as follow

$$K \triangleq \text{card}(\hat{\mathbf{Y}}) < \frac{\mu + 1}{2\mu}. \quad (3)$$

This condition requires that the host image must be sparse enough, which is impossible for natural image. Hua et al. [24] proposed that for RDH this condition in (3) can be relaxed as

$$K < M. \quad (4)$$

In the next section, we will propose how to maximize the embedding capacity by employing this condition above.

3. Description of Our Proposed Scheme

3.1. The Embedding Process. This section presents the proposed embedding algorithm, whose flowchart is shown in Figure 1. As in Figure 1, the embedding algorithm consists of four phases. In the first phase, a plain image can be encrypted by any of the existing cryptosystem. Next, the host image is represented sparsely by a given dictionary. Then, the encrypted plain image is embedded into the sparse coefficients of the host image. At last, the coefficients with message multiply with the new dictionary to generate the final cipher image, which is a meaningful image with visual security.

Step 1 (encrypt the plain image). Let a plain image and its cipher image in the first phase be $\mathbf{P} = \{P_{ij}\}_{i=1, j=1}^{m,n}$ and $\mathbf{C} = \{C_{ij}\}_{i=1, j=1}^{m,n}$. And the cipher image $\mathbf{C}$ is called Cipher image 1 in this paper. The plain image can be encrypted by any existing cryptosystem, whose keys are denoted by $A_1, A_2, \dots, A_T$ in Figure 1. Then we can obtain a vector image $\mathbf{c} \in \mathbb{R}^{mn \times 1}$ of $\mathbf{C}$.

Step 2 (represent the host image sparsely). To compress the host image, we must choose a dictionary $\mathbf{D}$ such as DCT dictionary, wavelet dictionary, noiselets dictionary and so on, which can also be obtained from dictionary learning. Let $\mathbf{H} \in \mathbb{R}^{M \times N}$ be a host image, whose vectorized version is $\mathbf{h}$

FIGURE 1: The flowchart of the proposed embedding method.

of length MN . The host vector $\mathbf{h}$ can be represented sparsely by $\mathbf{D} \triangleq [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{MN}] \in \mathbb{R}^{MN \times MN}$ as

$$\begin{aligned} \dot{\mathbf{Y}} &= \arg \min_{\mathbf{Y}} \|\mathbf{Y}\|_0 \\ \text{s.t. } \mathbf{h} &= \mathbf{D}\mathbf{Y}, \end{aligned} \quad (5)$$

where $\dot{\mathbf{Y}} \triangleq [y_1, y_2, \dots, y_{MN}]^T \in \mathbb{R}^{MN \times 1}$. Let the sparsity of $\mathbf{h}$ be K , then $\dot{\mathbf{Y}}$ can be obtained by orthogonal matching pursuit algorithm with the fixed sparsity K . Therefore, the vector $\dot{\mathbf{Y}}$ would have K nonzero elements. The indices of those nonzero elements are denoted by $\mathbf{k} \triangleq [k_1, k_2, \dots, k_K]$. If selecting the K nonzero elements in order, we can get a vector $\ddot{\mathbf{Y}} \triangleq [\dot{y}_{k_1}, \dot{y}_{k_2}, \dots, \dot{y}_{k_K}]^T \in \mathbb{R}^{K \times 1}$ satisfying

$$\dot{\mathbf{h}} = \Phi \ddot{\mathbf{Y}}, \quad (6)$$

where $\Phi \triangleq [\mathbf{d}_{k_1}, \mathbf{d}_{k_2}, \dots, \mathbf{d}_{k_K}] \in \mathbb{R}^{MN \times K}$ and $\dot{\mathbf{h}}$ is a lossy equivalent of $\mathbf{h}$.

Step 3 (embed the message into the coefficients vector of host image). Let the null space of Φ^T be $\Delta \in \mathbb{R}^{MN \times (MN-K)}$, which satisfying $\Phi^T \Delta = 0$ and $\Delta^T \Delta = \mathbf{I}$. We can first construct a new dictionary $\tilde{\mathbf{D}} = [\Phi, \Delta] \in \mathbb{R}^{MN \times MN}$. Then the Cipher image 1 is embedded in coefficients vector, which is depicted by Algorithm 1.

In Algorithm 1, the function “dec2bin()” converts the decimal number to a binary number and the function “bin2dec()” is opposite. The function “getbits($\mathbf{c}_b, i : j$)” gets the bits of $\mathbf{c}_b$ from the j th bit to the i th bit and constructs a binary number. The typical workflows are introduced as follows. First, by Line (4), each pixel of Cipher image 1 is represented by 8-bit binary number. Second, by Line (5)–(7), each 8-bit binary number is broken up into three parts and transformed to three decimal numbers $\mathbf{c}_3$, $\mathbf{c}_2$ and $\mathbf{c}_1$. These decimal numbers should be small enough to obtain a meaningful cipher image but large enough to eliminate the effect of quantization. Third, we propose parity checks

```

(1) Input: the Cipher image 1  $\mathbf{c}$ , sparse coefficients  $\ddot{\mathbf{Y}}$ 
(2)  $\mathbf{c}_3 = []$ ;  $\mathbf{c}_2 = []$ ;  $\mathbf{c}_1 = []$ ;
(3) for  $i = 1 : mn$ 
(4)    $\mathbf{c}_b = \text{dec2bin}(\mathbf{c}(i))$ ;
(5)    $s_3 = \text{bin2dec}(\text{getbits}(\mathbf{c}_b, 8 : 7))$ ;
(6)    $s_2 = \text{bin2dec}(\text{getbits}(\mathbf{c}_b, 6 : 4))$ ;
(7)    $s_1 = \text{bin2dec}(\text{getbits}(\mathbf{c}_b, 3 : 1))$ ;
(8)   for  $j = 1 : 3$ 
(9)      $s_j = s_j + 1$ ;
(10)    if  $(\text{mod}(s_j, 2) == 1)$ 
(11)       $s_j = s_j * (-1)$ ;
(12)    end
(13)  end
(14)   $\mathbf{c}_3 = [\mathbf{c}_3; s_3]$ ;  $\mathbf{c}_2 = [\mathbf{c}_2; s_2]$ ;  $\mathbf{c}_1 = [\mathbf{c}_1; s_1]$ ;
(15) end
(16) Output:  $\tilde{\mathbf{Y}} = [\ddot{\mathbf{Y}}, \mathbf{c}_3, \mathbf{c}_2, \mathbf{c}_1]$ 
  
```

ALGORITHM 1: Embed the Cipher image 1 into coefficients vector.

(Line (10)–(12)) to reduce the quantization error, where even numbers are set to be positive and odd numbers are negative. What’s more, due to the fact that zero is too small to reverse the positive and negative, we add 1 to s_j (Line (9)) to avoid the appearance of zero. Last, each group of 8-bit of Cipher image 1 is assigned to three locations of $\tilde{\mathbf{Y}}$ for output (Line (16)). Meanwhile, the length mn of $\mathbf{c}$ must equal to $(MN - K)/3$.

Step 4 (generate a meaningful image to transmit). A vector $\mathbf{x} \in \mathbb{R}^{MN \times 1}$ can be generated by the following equation

$$\mathbf{x} = \tilde{\mathbf{D}} \tilde{\mathbf{Y}} = [\Phi, \Delta] [\ddot{\mathbf{Y}}, \mathbf{c}_3, \mathbf{c}_2, \mathbf{c}_1]^T = \Phi \ddot{\mathbf{Y}} + \Delta \tilde{\mathbf{c}}, \quad (7)$$

where $\tilde{\mathbf{c}} = [\mathbf{c}_3, \mathbf{c}_2, \mathbf{c}_1]^T \in \mathbb{R}^{(MN-K) \times 1}$.

The meaningful image $\mathbf{X}$ to transmit can be obtained by reshape the vector $\mathbf{x}$. Moreover, because the elements in

FIGURE 2: The flowchart of the proposed extraction and reconstruction method.

vector $\tilde{\mathbf{c}}$ are much smaller than the minimum value of $\tilde{\mathbf{Y}}$, it has only a small effect on the transmitted image

$$\mathbf{x} \doteq \Phi \tilde{\mathbf{Y}} = \mathbf{h}. \quad (8)$$

The image quality to transmit depends on the sparsity K . The bigger the sparsity K , the nearer the image approximates to the host image.

3.2. The Extraction and Reconstruction Process. This section presents the proposed extraction and reconstruction algorithm, which is shown in Figure 2. As shown in Figure 2, this algorithm includes two phases. The first phase is to obtain the plain image by extracting the Cipher image 1 and decrypting it. The second phase is to recover the host image.

Step 1 (generate the matrix Φ and its null space Δ). Based on the vector $\mathbf{k}$ and dictionary $\mathbf{D}$, we can construct the matrix $\Phi \triangleq [\mathbf{d}_{k_1}, \mathbf{d}_{k_2}, \dots, \mathbf{d}_{k_K}]$ and compute the null space matrix Δ .

Step 2 (extract the plain image).

Sub-Step 1. Obtain the Cipher image 1 $\hat{\mathbf{c}}$ by the following Equation

$$\hat{\mathbf{c}} = \Delta^T \mathbf{x}_r = \Delta^T (\Phi \tilde{\mathbf{Y}} + \Delta \tilde{\mathbf{c}}) = \Delta^T \Delta \tilde{\mathbf{c}} = \tilde{\mathbf{c}}, \quad (9)$$

where $\mathbf{x}_r$ is the received cipher image vector. In application, quantification causes that the difference between vectors $\hat{\mathbf{c}}$ and $\tilde{\mathbf{c}}$ is 1. But, this quantification error can be eliminated by our parity checks strategy, which can have a great promoting effect on recovering the embedded image exactly. The extraction flows of the Cipher image 1 are shown in Algorithm 2.

Algorithm 2 achieves the recovery of $\tilde{\mathbf{c}}$ precisely, which is the inverse of Algorithm 1. In Algorithm 2, the function “fix()” takes out the integer part of a number, the function “sign()” takes out a positive or negative sign of a number and the function “strcat()” concatenates strings. The main flows are discussed as follows. First, by Line (4)–(16), the sign of $\tilde{\mathbf{c}}$ is extracted and its parity is determined to reduce the quantization errors, which correspond to Line (10)–(12) of

```

(1) Input: the received Cipher image  $\mathbf{x}_r$ , the sparsity  $K$ 
(2) Generate  $\Phi$  and  $\Delta$ 
(3)  $\Delta^T \mathbf{x}_r = \Delta^T (\Phi \tilde{\mathbf{Y}} + \Delta \tilde{\mathbf{c}}) = \Delta^T \Delta \tilde{\mathbf{c}} = \tilde{\mathbf{c}}$ 
(4) for  $i = 1 : mn - K$ 
(5)      $v = \text{fix}(\tilde{\mathbf{c}}(i));$ 
(6)      $s = \text{sign}(\tilde{\mathbf{c}}(i));$ 
(7)     if  $(\text{mod}(v, 2) == 0)$ 
(8)          $sv = 1;$ 
(9)     else
(10)          $sv = -1;$ 
(11)     end
(12)     if  $(s * sv == -1)$ 
(13)          $v = v + 1;$ 
(14)     end
(15)      $\tilde{\tilde{\mathbf{c}}}(i) = v;$ 
(16) end
(17)  $\tilde{\tilde{\mathbf{c}}} = \tilde{\tilde{\mathbf{c}}} - 1$ 
(18)  $\mathbf{c}_3 = \tilde{\tilde{\mathbf{c}}}(1 : (MN - K)/3)$ 
(19)  $\mathbf{c}_2 = \tilde{\tilde{\mathbf{c}}}((MN - K)/3 + 1 : (MN - K)/3 * 2)$ 
(20)  $\mathbf{c}_1 = \tilde{\tilde{\mathbf{c}}}((MN - K)/3 * 2 + 1 : MN - K)$ 
(21) for  $i = 1 : (MN - K)/3$ 
(22)      $s3 = \text{dec2bin}(\mathbf{c}_3(i));$ 
(23)      $s2 = \text{dec2bin}(\mathbf{c}_2(i));$ 
(24)      $s1 = \text{dec2bin}(\mathbf{c}_1(i));$ 
(25)      $s = \text{bin2dec}(\text{strcat}(s3, s2, s1));$ 
(26)      $\tilde{\mathbf{c}}(i) = s;$ 
(27) end
(28) Output:  $\hat{\mathbf{c}}$  and  $\tilde{\mathbf{c}}$ 

```

ALGORITHM 2: Extract the Cipher image 1.

Algorithm 1. Second, by Line (17), we have all elements of $\tilde{\tilde{\mathbf{c}}}$ minus 1, which correspond Line (9) of Algorithm 1. Third, by Line (18)–(20), $\tilde{\tilde{\mathbf{c}}}$ is divided into three parts (high, middle and low bit vectors), which correspond to Line (4)–(7) of Algorithm 1. Last, by Line (21)–(27), each cipher pixel is extracted, which correspond to Line (4)–(7) of Algorithm 1.

Sub-Step 2. Decrypt the Cipher image 1 $\hat{\mathbf{c}}$ and obtain the plain image.

FIGURE 3: Host images and encryption and decryption results of the plain image Lady: (a) host image Baboon; (b) host image Lena; (c) plain image Lady; (d) Cipher image 1 of (c); (e) the extracted image; (f) the extracted image.

Step 3 (recover the host image).

$$\hat{\mathbf{h}} = \mathbf{x}_r - \Delta \tilde{\mathbf{c}} = \Phi \tilde{\mathbf{Y}} = \hat{\mathbf{h}}. \quad (10)$$

4. Simulation Results

In this section, we verify our proposed embedding and extraction algorithm by experimental results. Without loss of generality, all host images and embedded images are with size 512×512 and 256×256 , respectively. Diaconu's encryption algorithm [29] is chosen to encrypt the plain image. The Dictionary we choose is DCT dictionary.

4.1. The Performance of Extraction and Reconstruction. We choose two host images (Baboon and Lena) and one plain image (Lady) to discuss, which are shown as Figures 3(a), 3(b) and 3(c). Figure 3(c) is encrypted by Diaconu's encryption algorithm [29] and the Cipher image 1 is shown in Figure 3(d). Then, Figure 3(d) is embedded in Figures 3(a) and 3(b) respectively. The embedding results are two cipher images Figures 4(a) and 5(a), which are visually secure. Figures 4(b) and 5(b) are the recovered host images from Figures 4(a) and 5(a). Only through our visual sense, we cannot distinguish the differences among the host image, the cipher image and the recovered host image.

Figure 4(c) is the difference image between Figures 3(a) and 4(b), and Figure 5(c) is the difference image between Figures 3(b) and 5(b). From Figures 4(c) and 5(c), we can conclude that there is very tiny difference between the

original host image and its recovered version. One of the applications of RDH is covertly communication, which aims to protect the embedded secret image, then, little distortion of the host image is acceptable in this scenario. To discriminate more about the differences among the host image, the cipher image and the recovered one, we draw their histograms. Figures 4(d)-4(e) are the histograms of Figures 3(a), 4(a) and 4(b) and Figures 5(d)-5(e) are the histograms of Figures 3(b), 5(a) and 5(b). Because Lena has more grey center of clustering than Baboon, its histograms Figures 5(d)-5(e) are even more telling. From Figures 5(d)-5(e), especially the the labeled regions, we can see that the recovered image is closer to the original host image than the cipher image with message, which verifies the performance of our proposed algorithm to recover the host image.

The plain images embedded in two host images can be extracted precisely, which are shown in Figures 3(e) and 3(f). One can see that the extracted image is exactly the same as Figure 3(a), indicating that the secret image can be covertly transmitted without any distortion.

4.2. The Embedding Rate. From Algorithms 1 and 2, the embedding rate can be denoted by

$$er = \frac{8}{3} \frac{MN - K}{MN}. \quad (11)$$

We explain the concluded embedding rate as follows. Supposing that a host image has MN pixels and K sparsity, there are

FIGURE 4: The cipher image Baboon and its reconstruction: (a) the cipher image containing message; (b) the recovered image; (c) the difference image between the original and recovered host image; (d-f) the histograms of Figures 3(a), 4(a) and 4(b).

$MN - K$ locations in coefficients vector $\tilde{Y}$ to embed message. The length mn of Cipher image 1 must equal to $(MN - K)/3$ as shown in Algorithm 1. Considering that each pixel is 8-bit, the Eq. (11) can be obtained.

By our thoroughly experiments, the embedding rate 2 bpp is a proper rate to balance the embedding capacity and reconstruction performance. If we want to achieve 2bpp embedding rate, the compressive ratio $K/MN = 0.25$. In our experiments, for the host image sized 512×512 and $er = 2$ bpp, there are $MN - K = 3MN/4$ locations to embed message. For each pixel of Cipher image 1 is assigned to 3 locations, the total pixels of additional plain image should be $MN/4$. Therefore, we choose a plain image sized 256×256 to verify our algorithm as shown in Figure 3.

If we want to improve the quality of the recovered image, the parameter K can be increased. Two indexes, Peak signal to Noise Ratio (PSNR) and Structure Similarity (SSIM), are suitable to measure the quality of the recovered host image, which are shown in Eqs. (12) and (13)

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2 M^2 N^2}{\sum_{i=1}^{M^2 N^2} (H_{oi} - H_{ri})^2} \right), \quad (12)$$

where H_{oi} and H_{ri} denote the i th pixels of the original and recovered host images respectively

$$\text{SSIM} = \frac{(2\mu_x \mu_y + C_1)(2\sigma_x \sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (13)$$

where μ_x and μ_y are mean values of two host images, σ_x and σ_y are standard deviations, C_1 and C_2 are two adjust parameters. For PSNR, a higher value is better; For SSIM, the nearer a value approximates to 1, the better the recovered image is.

Figure 6 shows that the PSNRs of the recovered host image increase with the raising of compressive ratio ($K/(MN)$) for three test host images, i.e., Lena, Pepper and Baboon. Meanwhile, different images have different reconstruction performances by using a fixed dictionary. For example, the smooth image Lena is more suitable for embedding data while the textured image Baboon is not. Table 1 gives the SSIMs of the recovered host images with the compressive ratios ranging in $[0.25, 0.75]$, where three images Lena, Pepper and Baoon are chosen as host images. From Table 1, one can see that the SSIMs are all close to 1, which show our proposed algorithm has good performance to recover host image. Since the closer to 1 the SSIM the better the reconstruction is, based on the Table 1 we can see Baboon

FIGURE 5: The cipher image Lena and its reconstruction: (a) the cipher image containing message; (b) the recovered image; (c) the difference image between the original and recovered host image; (d–f) the histograms of Figures 3(b), 5(a) and 5(b).

FIGURE 6: The PSNR VS the compressive ratio for three host images.

is the least suited to embedding data and Lena is the most suited one. The analyses from Table 1 are consistent with those from Figure 6.

Because the compressive ratio is proportional to the sparsity K and the embedding rate decreases with the increasing the sparsity K , the embedding rate decreases with the increasing the compressive ratio.

4.3. Against Noises. Each pixel of the embedded Cipher image 1 is transformed into 8 bits. These bits are divided into three parts and included in the coefficients vector $\tilde{Y}$ by Line (4)–(7) of Algorithm 1. Therefore, these coefficients related to the embedded pixels are usually smaller than those of host image. If the cipher image is exposure in noisy environment during transmission, the smaller coefficients would be influenced seriously. Therefore, the embedded Cipher image 1 is more sensitive to noises than the host image. Figure 7 discusses the sensitivity of the proposed algorithm to the noises. We add two levels “salt & pepper” noises to the received images. Figures 7(a) and 7(d) are the received cipher images, and they are full of “salt & pepper” noises. Figure 7(a) has noise density of 0.001 and Figure 7(d) is with 0.01. Figures 7(b) and 7(e) are the recovered host images, which are closer to the original host image. However, the quality of the extracted images are not satisfactory, as shown in Figures 7(c) and 7(f), which are in the presence of noises. These results manifest that the proposed algorithm has good reconstruction performance in noisy environment, but the embedded image is sensitive to noises, and this property can be used to detect the potential attacks which is un conspicuous on the stego-image.

4.4. Complexity. We would present the complexity from two phases: embedding phase, extraction and reconstruction

TABLE 1: The SSIM VS. compressive ratio for three host images.

Image	Compressive ratio										
	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
Lena	0.9912	0.9939	0.9959	0.9972	0.9980	0.9987	0.9990	0.9993	0.9995	0.9997	0.9999
Pepper	0.9868	0.9907	0.9935	0.9954	0.9968	0.9978	0.9984	0.9989	0.9992	0.9995	0.9997
Baboon	0.9778	0.9845	0.9894	0.9926	0.9948	0.9964	0.9975	0.9983	0.9989	0.9993	0.9995

FIGURE 7: Lena and its extracted data in noisy environment: (a) the cipher images with density of 0.001 “salt & pepper” noises; (b) the recovered host image from (a); (c) the extracted plain image from (a); (d) the cipher images with density of 0.01 “salt & pepper” noises; (e) the recovered host image from (d); (f) the extracted plain image from (d).

phase. In embedding phase, the complexity of encrypting plain image is related to the size of plain image. Therefore, this complexity is $O(m \times n)$. As for embedding the Cipher image 1, the complexities are related to the bits of the plain image and the sparse presentation by OMP. Then, the complexities are $O(8 \times M \times N)$ and $O(K \times M \times N)$ respectively, where K is sparsity. In summary, the complexity in the embedding phase is close to $O(K \times M \times N)$.

In extraction and reconstruction phase, the computational complexity can be effectively decreased due to the fact that matrix multiply instead of the CS recovery is executed in this phase, which is one of the main merits of this paper.

Then, the complexity in extraction and reconstruction phase is related to multiplication between Δ and the cipher image, which is close to $O(M \times N \times (MN - K))$. Therefore the total complexity of our algorithm is $O(M^2 \times N^2)$.

5. Comparison and Discussion

In what follows, we compare our proposed algorithm with three CS based RDH methods [22–24].

First, we discuss these algorithms from the view of the embedded image. Our proposed algorithm embeds natural image, but Refs. [22–24] embed binary image. Binary image

FIGURE 8: Lena and its extracted data in Gaussian noisy environment: (a) the noisy cipher image containing message; (b) the recovered host image; (c) the extracted image.

is much sparser than natural image and it is much easier to be embedded.

Second, from the view of embedding location, we discuss their differences. In [22], the message is embedded in the AC coefficients, and then is compressed by CS. If one wants to extract the message precisely, the plain image must have ultra-sparse pixels. Yamaç et al. [23] proposed a method to embed data in the measurements, but our method is to embed the data in the sparse coefficients. The embedded locations in Hua et al. [24] are the same as those of our proposed algorithm, but there is drastically difference between them.

In [24], Hua et al. expanded possible locations to embed data, and they also dictated that there are $MN - K$ locations for data hiding except the coefficients of host image. At the same time they indicated that multiple chips can be inserted, but the length of multiple chips was not stated explicitly. If inserting multiple chips at the same time, using their detection algorithm multiple chips are inseparable and cannot be extracted precisely. Therefore, they did not give the method how to embed and extract multiple bits. In a word, our proposed work improves Hua's algorithm in two ways. One is to give a method to embed data into $MN - K$ locations at the same time and to extract them precisely. The other is to insert more than two bits into one location.

Last, from the view of embedding rate, our proposed work achieves the largest rate. As previously described, our embedding rate achieves 2 bpp, which is larger than those in [22–24], but it leads to the sensitivity to noises. If we insert

one bit into a coefficient location, the demerit (sensitivity to noises) can be made up. The idea is discussed as follows.

In Line (16) of Algorithm 1, the output can be changed to $\tilde{\mathbf{Y}} = [\tilde{\mathbf{Y}}, \alpha \mathbf{b}]$, where α denotes the embed strength and $\mathbf{b} = \{b_i\}_{i=1}^{MN-K}$ represents the embedded bits. If the i th embedded data is 0, then $b_i = 1$. If the i th embedded data is 1, then $b_i = -1$. Thus, the embed rate can achieve $(MN - K)/MN$. If the transmitted cipher image is polluted by noises, the receiver data is changed to

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{z} = \Phi \tilde{\mathbf{Y}} + \Delta \tilde{\mathbf{c}} + \mathbf{z}, \quad (14)$$

where $\mathbf{z}$ is additive noise satisfying $\|\mathbf{z}\|_2 \leq \epsilon$. After extracting data like Line (3) of Algorithm 2, we can obtain

$$\Delta^T \tilde{\mathbf{x}} = \Delta^T (\mathbf{x} + \mathbf{z}) = \Delta^T (\Phi \tilde{\mathbf{Y}} + \Delta \tilde{\mathbf{c}} + \mathbf{z}) = \tilde{\mathbf{c}} + \Delta^T \mathbf{z}. \quad (15)$$

For $\|\Delta^T \mathbf{z}\|_2 = (\Delta^T \mathbf{z})^T (\Delta^T \mathbf{z}) = \mathbf{z}^T \Delta \Delta^T \mathbf{z} = \mathbf{z}^T \mathbf{z} = \|\mathbf{z}\|_2 \leq \epsilon$, we can change (15) into

$$\Delta^T \tilde{\mathbf{x}} = \tilde{\mathbf{c}} + \tilde{\mathbf{z}} = \alpha \mathbf{b} + \tilde{\mathbf{z}}, \quad (16)$$

where $\tilde{\mathbf{z}} = \Delta^T \mathbf{z}$.

When extracting, we can use $\alpha \text{sign}(\Delta^T \tilde{\mathbf{x}}) = \alpha \text{sign}(\tilde{\mathbf{c}} + \tilde{\mathbf{z}}) = \alpha \text{sign}(\alpha \mathbf{b} + \tilde{\mathbf{z}})$ to replace Line (5) of the Algorithm 2. If $\|\tilde{\mathbf{z}}\|_\infty < \alpha$, then $\alpha \text{sign}(\Delta^T \tilde{\mathbf{x}}) = \alpha \mathbf{b}$, i.e., the embedded bits can be extracted precisely. In simulation, α is set to be 5.

The simulation results are shown from Figures 8–10. Figures (a) are the transmitted cipher images. Because each

FIGURE 9: Lena and its extracted data in Gaussian noisy environment: (a) the noisy cipher image containing message; (b) the recovered host image; (c) the extracted image.

FIGURE 10: Lena and its extracted data in "salt & pepper" noisy environment: (a) the noisy cipher image containing message; (b) the recovered host image; (c) the extracted image.

pixel in a binary image can be represented by 1 bit, we embed a binary image Lady in three Figures (a). Figure 8(a) is polluted by Gaussian noise, where noise level is smaller than the embed strength α . From Figure 8(c), we can see the extracted image is without noises. In contrast, the Gaussian noise level in Figure 9(a) is larger than the embed strength and from Figure 9(c) we can see the extracted image is degraded by Gaussian noise. In Figure 10(a), the cipher image is exposed in “salt & pepper” noisy environment. From Figure 10(c), we can see that the extracted result is degraded as same as Figure 9(c). It is because the “salt” noises are much larger than α , and the “pepper” noises cleared some information of transmitted data. In fact, these noises caused by strong Gaussian or “salt & pepper” cannot be eliminated by these methods [22–24].

We can conclude, for our proposed algorithm in this section, decreasing the embedding rate and embedding a bit in a location of coefficients vector can improve the robustness to noises. If the compressive ratio is still $K/MN = 0.25$, the embedding rate is 0.75 bpp, which is still larger than those in [22–24].

6. Conclusions

In this paper, an algorithm for meaningful image encryption is proposed. The main idea is to embed the secret image into the sparse representation of host image to achieve high covertness. Our proposed algorithm has the following merits. First, a novel method of RDH in CS domain is proposed, and the data embedding capacity is significantly higher than other similar schemes. Second, we propose the parity checks strategy to eliminate quantization error. Third, if the transmitted cipher image is not polluted by noises, the embedding rate achieves 2 bpp. And, if the transmitted cipher image is polluted by additive noise with lower level, the embedding rate $(MN - K)/(MN)$ bpp is equal to the difference between 1 and compressive ratio. Last, we use matrix multiplication in reconstruction phase to decrease computational complexity. In the future, we aim to improve the robustness of the proposed algorithm with lower computational complexity.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61602158, 61572089, 61633005, U1604156, 61671042, 61403016), the China Post-doctoral Science Foundation (Grant no. 2016M600030), the Beijing Natural Science Foundation (Grant no. 4172037), the Open Fund Project of Fujian Provincial Key Laboratory in Minjiang University (Grant no. MJUKF201702), the Science Foundation for the Excellent Youth Scholars of Henan Normal University (Grant no. YQ201607), the Natural Science Foundation of Chongqing Science and Technology Commission (Grant no. cstc2017jcyjBX0008), and the Fundamental

Research Funds for the Central Universities (Grants nos., 106112017CDJQJ188830, 106112017CDJXY180005).

References

- [1] X. Liao, J. Yin, and S. Guo, “Medical JPEG image steganography based on preserving inter-block dependencies,” *Computers Electrical Engineering*, 2017.
- [2] X. Liao, Z. Qin, and L. Ding, “Data embedding in digital images using critical functions,” *Signal Processing: Image Communication*, vol. 58, pp. 146–156, 2017.
- [3] M. Wu, H. Yu, and B. Liu, “Data hiding in image and video: Part II - Designs and applications,” *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 696–705, 2003.
- [4] B. Feng, W. Lu, and W. Sun, “Secure binary image steganography based on minimizing the distortion on the texture,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, article no. A3, pp. 243–255, 2015.
- [5] X. Liao, G. Chen, and J. Yin, “Content-adaptive steganalysis for color images,” *Security and Communication Networks*, vol. 9, no. 18, pp. 5756–5763, 2016.
- [6] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, “Lossless generalized-LSB data embedding,” *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.
- [7] X. Liao, K. Li, and J. Yin, “Separable data hiding in encrypted image based on compressive sensing and discrete fourier transform,” *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 20739–20753, 2017.
- [8] G. Bhatnagar, “Robust covert communication using high capacity watermarking,” *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 3783–3807, 2017.
- [9] B. Ma and Y. Q. Shi, “A reversible data hiding scheme based on code division multiplexing,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1914–1927, 2016.
- [10] S. A. Parah, J. A. Sheikh, A. M. Hafiz, and G. M. Bhat, “A secure and robust information hiding technique for covert communication,” *International Journal of Electronics*, vol. 102, no. 8, pp. 1253–1266, 2015.
- [11] F. Chen, K.-w. Wong, X. Liao, and T. Xiang, “Period distribution of generalized discrete Arnold cat map,” *Theoretical Computer Science*, vol. 552, pp. 13–25, 2014.
- [12] Y. Zhang, J. Zhou, F. Chen et al., “Embedding cryptographic features in compressive sensing,” *Neurocomputing*, vol. 205, pp. 472–480, 2016.
- [13] Y. Zhang, J. Zhou, F. Chen et al., “A block compressive sensing based scalable encryption framework for protecting significant image regions,” *International Journal of Bifurcation and Chaos*, vol. 26, no. 11, Article ID 1650191, 2016.
- [14] A. Kanso and M. Ghebleh, “An algorithm for encryption of secret images into meaningful images,” *Optics and Lasers in Engineering*, vol. 90, pp. 196–208, 2017.
- [15] X. Chai, Z. Gan, Y. Chen, and Y. Zhang, “A visually secure image encryption scheme based on compressive sensing,” *Signal Processing*, vol. 134, pp. 35–51, 2017.
- [16] L. Kamstra and H. J. Heijmans, “Reversible data embedding into images using wavelet techniques and sorting,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2082–2090, 2005.
- [17] J. Tian, “Reversible data embedding using a difference expansion,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

- [18] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication and Image Representation*, vol. 28, pp. 21–27, 2015.
- [19] M. Li and Y. Li, "Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding," *Signal Processing*, vol. 130, pp. 190–196, 2017.
- [20] M. Li, D. Xiao, Y. Zhang, and H. Nan, "Reversible data hiding in encrypted images using cross division and additive homomorphism," *Signal Processing: Image Communication*, vol. 39, pp. 234–248, 2015.
- [21] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.
- [22] D. Xiao, H. Cai, Y. Wang, and S. Bai, "High-capacity separable data hiding in encrypted image based on compressive sensing," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13779–13789, 2016.
- [23] M. Yamaç, Ç. Dikici, and B. Sankur, "Hiding data in compressive sensed measurements: A conditionally reversible data hiding scheme for compressively sensed measurements," *Digital Signal Processing*, vol. 48, pp. 188–200, 2016.
- [24] G. Hua, Y. Xiang, and G. Bi, "When compressive sensing meets data hiding," *IEEE Signal Processing Letters*, vol. 23, no. 4, pp. 473–477, 2016.
- [25] M. Li, D. Xiao, and Y. Zhang, "Reversible data hiding in block compressed sensing images," *ETRI Journal*, vol. 38, no. 1, pp. 159–163, 2016.
- [26] Y. Rachlin and R. D. Baron, "The secrecy of compressed sensing measurements," in *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 813–817, IEEE, Urbana, Ill, USA, September 2008.
- [27] A. Orsdemir, H. O. Altun, G. Sharma, and M. F. Bocko, "On the security and robustness of encryption via compressed sensing," in *Proceedings of the IEEE Military Communications Conference (MILCOM '08)*, pp. 1–7, San Diego, Calif, USA, November 2008.
- [28] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [29] A.-V. Diaconu, "Circular inter-intra pixels bit-level permutation and chaos-based image encryption," *Information Sciences*, vol. 355–356, pp. 314–327, 2016.