Computers, Materials & Continua CMC, vol.65, no.3, pp.2623-2638, 2020

Reversible Data Hiding in Encrypted Images Based on Prediction and Adaptive Classification Scrambling

Lingfeng Qu[1], Hongjie He[1], Shanjun Zhang[2] and Fan Chen[1, *]

Abstract: Reversible data hiding in encrypted images (RDH-EI) technology is widely used in cloud storage for image privacy protection. In order to improve the embedding capacity of the RDH-EI algorithm and the security of the encrypted images, we proposed a reversible data hiding algorithm for encrypted images based on prediction and adaptive classification scrambling. First, the prediction error image is obtained by a novel prediction method before encryption. Then, the image pixel values are divided into two categories by the threshold range, which is selected adaptively according to the image content. Multiple high-significant bits of pixels within the threshold range are used for embedding data and pixel values outside the threshold range remain unchanged. The optimal threshold selected adaptively ensures the maximum embedding capacity of the algorithm. Moreover, the security of encrypted images can be improved by the combination of XOR encryption and classification scrambling encryption since the embedded data is independent of the pixel position. Experiment results demonstrate that the proposed method has higher embedding capacity compared with the current state-of-the-art methods for images with different texture complexity.

Keywords: Reversible data hiding, classification scrambling, prediction error, multi-bits embedding.

1 Introduction

Reversible data hiding (RDH) in images is a technique, by which the original cover can be losslessly recovered after the embedded secret information is extracted. Since marked images can be reconstructed without loss after embedding information, RDH technology is widely used in military, medical, and legal forensics applications [Jolfaei, Wu and Muthukkumarasamy (2016)]. In recent years, security issues in cloud computing have received widespread attention. Reliable storage and secure transmission of digital images has always been a concern for users. Uploading encrypted images to the cloud is an effective solution to protect data confidentiality and privacy. This is because the original, meaningful plaintext is converted to incomprehensible random noise by encryption

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu, 611756, China.

[2] Department of Information Science, The Faculty of Science, Kanagawa University, Kanagawa, 259129 Japan.

[*] Corresponding Author: Fan Chen. Email: fchen@swjtu.edu.cn.

Received: 16 January 2020; Accepted: 24 April 2020.

[Jolfaei, Wu and Muthukkumarasamy (2016)]. However, most existing studies of RDH are only appropriate for unencrypted images. Therefore, researchers have applied traditional reversible data hiding techniques to encrypted images, and many reversible data hiding in encrypted images (RDH-EI) methods have been developed. These RDH-EI methods can be classified into two categories: VRAE (Vacating Room After Encryption) and VRBE (Vacating Room Before Encryption) [Shi, Li, Zhang et al. (2016)].

The VRAE framework vacates the embedding room from the encrypted images directly, and the secret information are embedded in encrypted images [Yin, Luo and Hong (2014); Yin, Abel, Zhang et al. (2016); Fu, Kong, Yao et al. (2019); Zhou, Sun, Dong et al. (2016); Qian and Zhang (2016); Qin, He, Luo et al. (2018); Qin, Zhang, Cao et al. (2018); Liu and Pun (2018)]. Yin et al. [Yin, Luo and Hong (2014); Yin, Abel, Zhang et al. (2016)] have proposed a RDH-EI algorithm based on VRAE framework, the encrypted image is divided into non-overlapping blocks first, and then in each block the secret information is embedded by the histogram shift method, the maximum embedding rate is 0.1bpp. In order to increase the embedding capacity, the encrypted image is compression-encoded in Fu et al. [Fu, Kong, Yao et al. (2019); Zhou, Sun, Dong et al. (2016); Qian and Zhang (2016); Qin, He, Luo et al. (2018); Qin, Zhang, Cao et al. (2018); Liu and Pun (2018)]. For users, the VRAE algorithm is simple and efficient to operate, because VRAE only needs to encrypt the image without any extra preprocessing. However, generally speaking, due to the maximum entropy of the encrypted image, it is relatively difficult and inefficient to vacate space from the encrypted image.

To solve the problems mentioned above in VRAE, researchers have proposed the RDH-EI algorithm to vacant room before encryption [Ma, Zhang, Zhao et al. (2013); Xu and Wang (2016); Cao, Du, Wei et al. (2016); Yi and Zhou (2017); Pauline and William (2018)]. This may increase the burden on the content owner. However, preprocessing before encryption is worthwhile because VRBE can vacate more space to embed extra information since the correlation of the original image is used. In 2013, Ma et al. [Ma, Zhang, Zhao et al. (2013)] proposed the first VRBE algorithm, which uses the traditional RDH algorithm to embed some LSBs into the smoother pixels before using the stream cipher to encrypt the original image, so as to reserve space for information hiding. The information embedding rate in Ma's method [Ma, Zhang, Zhao et al. (2013)] is up to 0.5bpp. Xu et al. [Xu and Wang (2016)] proposed a RDH-EI by adopting the interpolation prediction errors coding combined with traditional RDH, in which a specific encryption mode was designed to encrypt the interpolation error. The maximum embedded capacity of the Xu's method [Xu and Wang (2016)] is 0.75 bpp. Both Ma et al. [Ma, Zhang, Zhao et al. (2013)] and Xu et al. [Xu and Wang (2016)] make room based on traditional RDH technology in encrypted images, and the algorithm has low time complexity. However, since the histogram shifting technique can only embed 1-bit information in one pixel, the embedding capacity of the above two algorithms is not high. To further improve embedding capacity, researchers have proposed RDH-EI based on compression and encoding. A patch-level sparse representation was adopted in Cao's method [Cao, Du, Wei et al. (2016)] to increase the embedding payload to close to 1 bpp. Yi et al. [Yi and Zhou (2017)] proposed a Binary-block embedding (BBE) RDH-EI algorithm, the embedding rate of BEE algorithm is 2 bpp. All the methods mentioned above need to compress and encode the image before encryption, in order to recover the

encrypted image losslessly, decoding is needed in the decryption process [Xiang, Wang, Yang et al. (2017); Xiang, Wu, Li et al. (2018)]. The operation of encoding and decoding results in high time complexity of the algorithm. To reduce the time complexity of the algorithm, the MSB-inversion prediction based RDH-EI methods [Pauline and William (2018)] have been proposed. The MSB method makes room for the most significant bit plane of the image based on prediction technology, the maximum embedding capacity is less than 1bpp and the algorithm time complexity is greatly reduced relative to compression and coding methods. On the other hand, in order to improve the security of the encrypted images, the pixel classification scrambling strategy have been proposed [Chen, Yin, He et al. (2018); Qu, He and Chen (2019); Das, Baykara and Tuna (2019)]. All of the above algorithms use the combination of scrambling and XOR encryption to improve the security of encrypted images. The encryption method combining XOR and scrambling has the advantages of simplicity and high efficiency.

In order to further improve the embedding capacity and security of RDH-EI algorithm, this paper proposed a reversible data hiding algorithm for encrypted images based on prediction and adaptive classification scrambling. Main contributions of this paper include the following aspects.

(i) A novel prediction error technique is adopted and the optimal threshold is selected adaptively. Data are embedded into multiple high significant bits of the pixel values within the optimal threshold to make more embedding space from the original image.

(ii) The classification scrambling encryption is used to improve the security performance of the encrypted image.

Experiment results show that the proposed RDH-EI method can make more embedding space than the latest methods [Yin, Luo and Hong (2014); Liu and Pun (2018); Xu and Wang (2016); Yi and Zhou (2017); Pauline and William (2018)]. The rest of this paper is organized as follows. The proposed RDH-EI method is described in detail in Section II. Experimental results and analysis are presented in Section III. Section IV is the conclusions.

2 Proposed scheme

In this section, a RDH method in encrypted version of images is illustrated, which is made up of image encryption, data hiding in encrypted image, data extraction and image recovery phases.

The content owner first reserves enough space on original image and then converts the image into its encrypted version with the encryption key. Then, content owner needs to upload the encrypted image to the cloud. Data hider in the cloud for easy management of encrypted images, he can embed some secret information into the encrypted image. When the receiver sends a request to the cloud, data hider can extract the hidden information to find the correct encrypted image and transmit it to the legitimate receiver. The legitimate receiver, maybe the content owner himself or an authorized third party, downloads the encrypted image from the cloud. Different from the traditional algorithm framework, in the proposed algorithm (as shown in Fig. 1), the receiver can choose to obtain the marked-encrypted image, or can choose to get the encrypted image without marking. This is because the secret information and the image information are separable in this algorithm. When the receiver needs to authenticate the encrypted image, he/she may need

the marked-encrypted image. If the receiver does not need the secret information hidden in the encrypted image, he/she can choose to download the encrypted image directly.

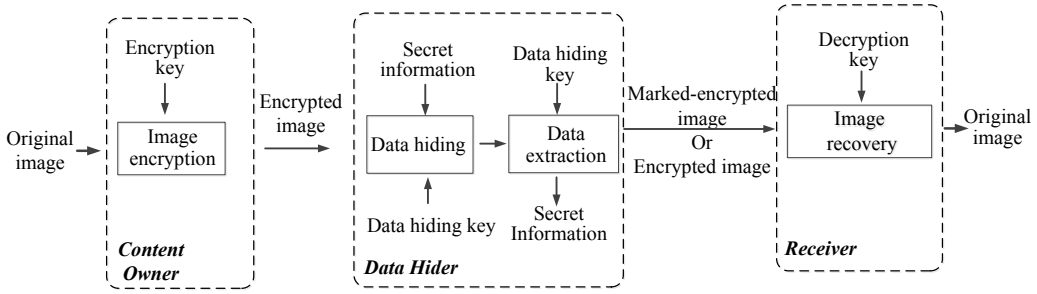


Figure 1: The framework of proposed scheme

2.1 Image encryption

In most of the existing RDH-EI algorithms based on prediction errors, the improvement of histogram shift technology is focused on. However, the visibility of images in the encrypted domain does not need to be considered, so we adopt a new strategy to increase the capacity after predicting the error. Since the prediction errors are mostly concentrated near 0, redundant multi-bit spaces can be used to make room in a prediction error values. In order to improve the security performance of the encrypted image under the high embedded capacity, the image is encrypted by the combination of bitwise XOR and scrambling to ensure the security of the encrypted images. Next, we will introduce each process in the proposed algorithm in details.

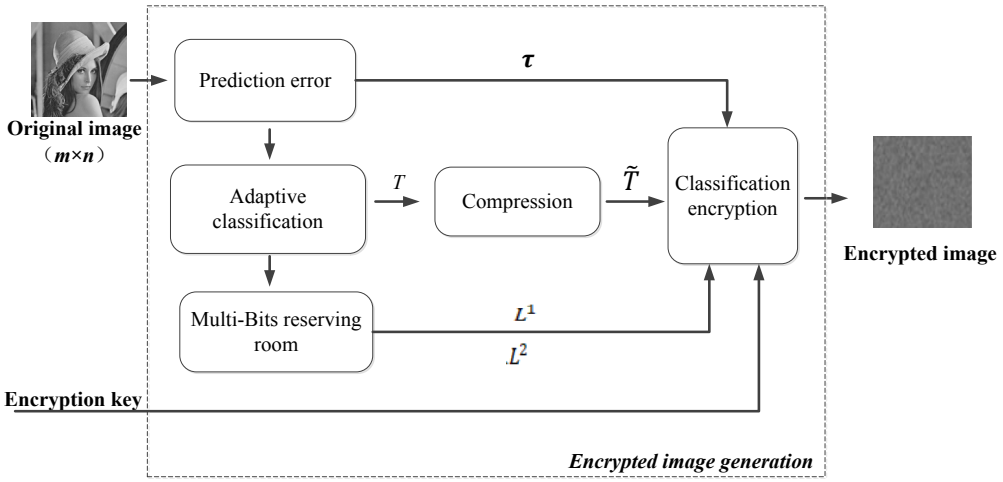


Figure 2: The framework for generating an encrypted image

The framework for generating an encrypted image is shown in Fig. 2. First of all, a novel prediction method is used to make room in plaintext image. Different from traditional preprocessing algorithms based on prediction errors, the proposed algorithm adaptively finds the optimal threshold after pixel prediction errors, and then vacates multi-bits of a

pixel to embed information. Compared the algorithms of Xu et al. [Xu and Wang (2016)]; Pauline and William (2018)], the proposed algorithm more efficiently utilizes pixel bits.

2.1.1 Prediction error

Assume the original image X is an 8 bit gray-scale image with its size $m \times n$ and pixels $x_{i,j} \in [0,255]$, $X = \{x_{i,j} | i = 1,2, \dots, m, j = 1,2, \dots, n\}$. The prediction error image X', $X' = \{x'_{i,j} | i = 1,2, \dots, m, j = 1,2, \dots, n\}$,is obtained by

$$x'_{i,j} = \begin{cases} x_{i,j} & i = 1, \quad j = 1 \\ x_{i,j} - x_{i,j-1} & i = 1, \ j = 2,3 \dots, n \\ x_{i,j} - x_{i-1,j} & i = 2,3 \dots, m, \ \ j = 1 \\ [\, x_{i,j} - \frac{x_{i-1,j-1} + x_{i-1,j} + x_{i,j-1}}{3} \,] & i = 2, \dots, m, \ \ j = 2, \dots, n \end{cases} \tag{1}$$

In Eq. (1), [.] represents the rounding function. Since the prediction error $x'_{i,j}$ can either be positive or negative, the problem of rep resenting a number's sign can be to allocate one sign bit to represent the sign: set that bit (often the most significant bit) to 1 for a positive number, and set to 0 for a negative number. The remaining bits in the number indicate the magnitude (or absolute value). Hence only 7 bits (apart from the sign bit) can be used to represent the magnitude which can range from 0000000 (0) to 1111111(127). Thus only $x'_{i,j}$ falling with the range of +127 and -127 can be stored in 8 bits once the sign bit (the eighth bit) is added. To recover the original image perfectly, prediction error $x'_{i,j}$ is truncated into [-127, +127] by Eq. (2). After the calculation of Eq. (2), the prediction error image is converted to X'', $X'' = \{x''_{i,j} | i = 1,2, \dots, m, j = 1,2, \dots, n\}$.

$$x''_{i,j} = \begin{cases} x'_{i,j} + 127 & x'_{i,j} < -127 \\ x'_{i,j} - 127 & x'_{i,j} > +127 \\ x'_{i,j} & -127 \le x'_{i,j} \le +127 \end{cases} \tag{2}$$

In order for reversibility, a binary array, i.e., location map 'τ' , is introduced to record the positions of the points, in which "1" for $x''_{i,j} \notin [-127, +127]$ and "0" for others. Most of the prediction error is within the range of [-127, 127] because of the local correlation of the image pixels. So the location map 'τ' is mainly composed of '0', which can be compressed to a small number of bits using the lossless compression algorithm, such as run length code.

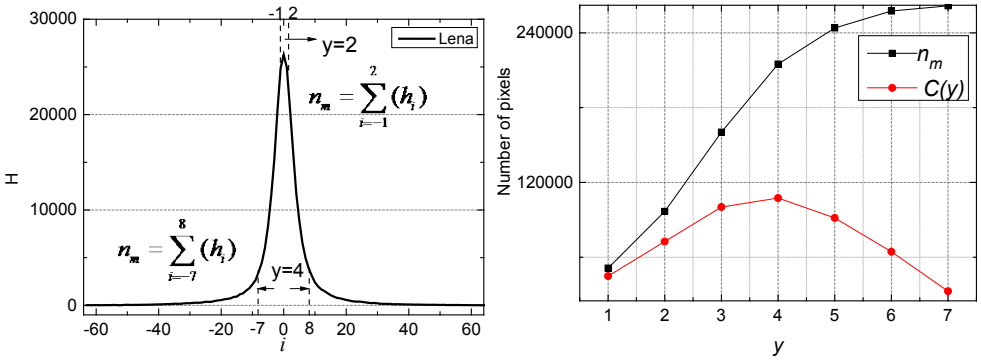
2.1.2 Adaptive classification

After the content owner gets X'', in order to get the highest embedded capacity and recover the original image without loss, the values in X'' needs to be classified according to the threshold y. An threshold range $[T_n, T_p]$ can be determined by the threshold y. All possible cases of T_n and T_p are:

$$\begin{cases} T_n = & -2^{y-1} + 1 \\ T_p = & 2^{y-1} \end{cases} \quad (y = 1,2, \dots 7) \tag{3}$$

Here, we can get seven different threshold ranges. The maximum threshold range is [-63,

64], the prediction error of 'Lena' within the maximum threshold range [-63,64] is shown in Fig. 3(a).



(a) Prediction error of Lena (b) $C(y)$ and n_m

Figure 3: The prediction error of 'Lena' within the maximum threshold range [-64, 64] and the relationship between $C(y)$ and n_m

In Fig. 3(a), when y is increased from 2 to 4, the threshold region is extended from [-1, 2] to [-7, 8], accordingly, the number of pixels n_m in the threshold region is also increased. However, the amount of vacated space is not only related to n_m. For the convenience of analysis, the space $C(y)$ to be vacated is expressed in the form of pixel, $C(y)$ can be calculated by:

$$C(y) = \frac{(8-y)\times n_m}{8} \quad (y = 1,2,\dots 7) \tag{4}$$

where y is a threshold with the range of [1, 7]. Define the prediction error histogram as H , $\mathrm{H} = \{h_i | i = -63,\cdots,0,\cdots,64\}$, $n_\mathrm{m}$is the number of pixels within the threshold range:

$$n_m = \sum_{i=T_n}^{T_p}(h_i) \qquad -63 \le T_n < T_p \le 64 \tag{5}$$

It can be seen from the Eq. (4) that y and n_m together determine the $C(y)$, so it is necessary to balance y and n_m to maximize the vacated space $C(y)$. The relationship between $C(y)$ and $n_\mathrm{m}$of 'Lena' is shown in Fig. 3(b). It can be intuitively found that $C(y)$ decreases as the n_m increases, so an optimum threshold y_m can be obtained when n_m is relatively large and y is small. In Fig. 3(b), the y value corresponding to the peak point of $C(y)$ is the optimal threshold. Obviously, the optimal threshold for 'Lena' is 4. In mathematics, y_m can be obtained by:

$$y_m = \underset{y}{\operatorname{argmax}}\ (C(y)) \quad (y = 1,2,\dots 7) \tag{6}$$

The threshold y_m is adaptively selected, and the selected threshold ensures the maximum embedding capacity. The classification matrix T determined by the optimal threshold y_m is:

$$T_{i,j} = \begin{cases} 1 & x''_{i,j} \in [-2^{y_m-1}+1, 2^{y_m-1}] \\ 0 & Others \end{cases}, \quad 1 \le i \le m, 1 \le j \le n \tag{7}$$

Note that the pixel classification matrix T is a binary matrix consists of 0 and 1.We modifies and simplifies the block classification and coding method of BBE (binary block

embedding) algorithm proposed by Yi et al. [Yi and Zhou (2017)], which makes it feasible to compress the *T* matrix. The main modifications to BEE algorithm are as follows:

(1) We divide the *T* matrix into multiple non-repeating 4×4 blocks.

(2) Modify the classification and encoding of blocks. Tab. 1 shows the classification of the block types of the *T* matrix.

Table 1: Block classification

Condition	Block type	Description	Block-labeling bits
$n0 = 0$	G-I	All pixels are 1	1
$1 \le f \le n_a, n0 < n1$	G-II	Most pixels are 1	01
Other	Bad	Other cases	00

2.1.3 Multi-bits reserving room and encryption

In the previous section, we chose the optimal threshold area for more embedded capacity. The optimal threshold range classifies the prediction error into two categories: the values within the threshold range and the values outside the threshold range. Through the pixel classification matrix, the position and type of the two categories can be recorded. In this way, the two types of pixels can perform different encoding operations. First, the value within the optimal threshold and the value outside the threshold constitute a one-dimensional sequence L^1 and L^2, which are obtained by

$$\begin{cases} L^1 = \{l_v | l_v \in [-2^{y_m-1}+1, 2^{y_m-1}]\,, v = 1,2,\dots,\ n_m\} \\ L^2 = \{l_u | l_u \in [-127, -2^{y_m}+1) \cup (2^{y_m}, +127]\,, u = 1,2,\dots, \mathrm{U}\} \end{cases} \tag{8}$$

where U= $m \times n - n_m$, both l_v and l_u belong to X''.The element l_v in L^1 is re-encoded to obtain l'_v, and the encoding method is as:

$$l'_v = l_v + |-2^{y_m-1}+1| \tag{9}$$

The encoded l'_{v} is all positive and only needs y_m bits to represent, then $8 - y_m$ bits are reserved room for embedding data. Difference outside the threshold range can either be positive or negative, the problem of representing a number's sign can be to allocate one sign bit to represent the sign: set that bit (often the most significant bit) to 1 for a positive number, and set to 0 for a negative number. Since all the values are normalized to [-127, +127] during the prediction error, the elements in L^2 do not overflow. The 8 bits of the element in L^2 need to be reserved and cannot be compressed. An example is given in Fig. 4. When y_m=2, the threshold range is [-1, 2]. The threshold range becomes [0, 3] which encoded by the Eq. (9). The encoded L^1 can be represented by 16 bits. Each pixel value of L^2 needs to be represented by 8 bits. The size of T matrix (uncompressed) is 4×4 bits, which is stored after L^2. There is no overflow value in Fig. 4, so τ is not stored. The elements in L^1 and L^2 are encrypted and reorganized by XOR and scrambling respectively, and the encrypted image E is generated, $E = \{e_{i,j} | i = 1,2,\dots,m, j = 1,2,\dots,n\}$.

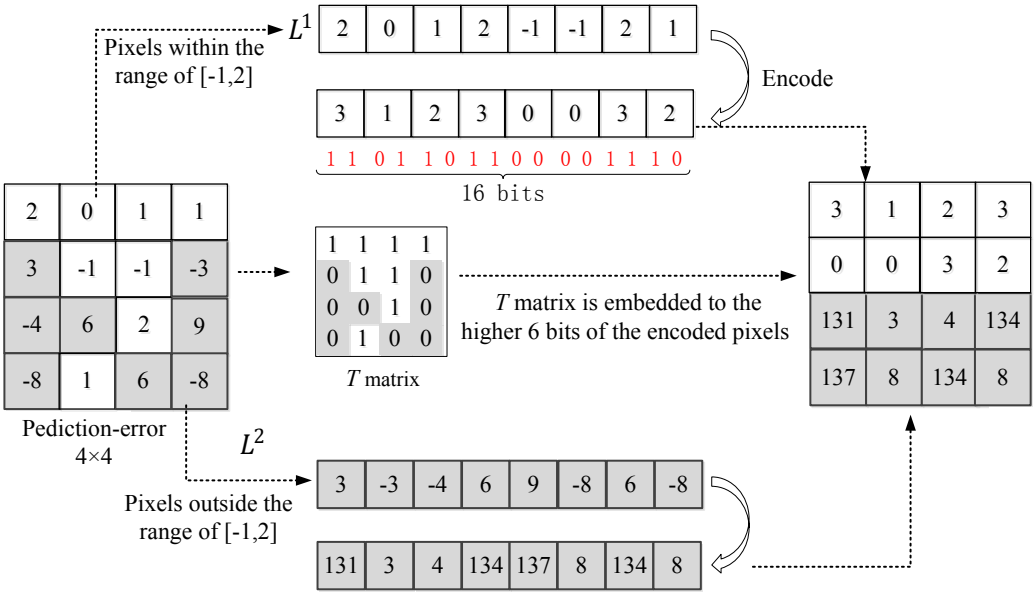


Figure 4: Multi-bits reserving room and embedding the *T* matrix

2.2 Data hiding and extraction

The content owner gets the encrypted image, uploading the encrypted image to the cloud. The cloud administrator needs to embed the secret information in the encrypted image for the convenience of management.

Cloud administrator first concatenates the encrypted images into a one-dimensional vector by column and converts all encrypted pixel values into a binary bit sequence stream. The 100-bit auxiliary information at the end of the binary bit stream of the encrypted image is converted to a decimal number, and then the reserved user-specified capacity can be obtained. Secret information is embedded in the encrypted bit stream by substitution. Eq. (10) describes the maximum embedding capacity.

$$C_m = (8 - y_m) \times n_m - \mathrm{L}_{\tilde{\mathrm{T}}} - \mathrm{L}_{\tau} - \mathrm{L}_{\mathrm{a}} \tag{10}$$

Here, L_{τ} is the length of the overflow map matrix, $\mathrm{L}_{\tilde{\mathrm{T}}}$ is the length of the compressed pixel classification matrix, L_{a} is the length of the auxiliary information and n_m is the number of values within the threshold range. $m \times n$ is the size of the original image. Equation for calculating the maximum embedding rate is shown in Eq. (11).

$$\text{Rate} = \frac{C_m}{m \times n} \quad \text{(bpp)} \tag{11}$$

The data extraction process is the inverse process of the embedding process. Cloud managers first get the auxiliary information y_m and the length information of L^1, and extract the secret information from the high $8 - y_m$ bits of L^1. The detailed process of data hiding and extraction can be referred to Qu et al. [Qu, He and Chen (2019)].

2.3 Image recovery

After extracting the secret information, the receiver no longer decrypts the encrypted image, but directly restores the original image. The recovery process of the element positions in L^1 and L^2 is the same as [Qu, He and Chen (2019)] and will not be described in detail here. The image after restoring the position is $X'', X'' = \{x''_{i,j} | i = 1,2, \dots, m, j = 1,2, \dots, n\}$. The overflow values is restored according to the map matrix. The recovery of the overflow values is shown in Eq. (12). After recovering the overflow values, we got the prediction error image X'.

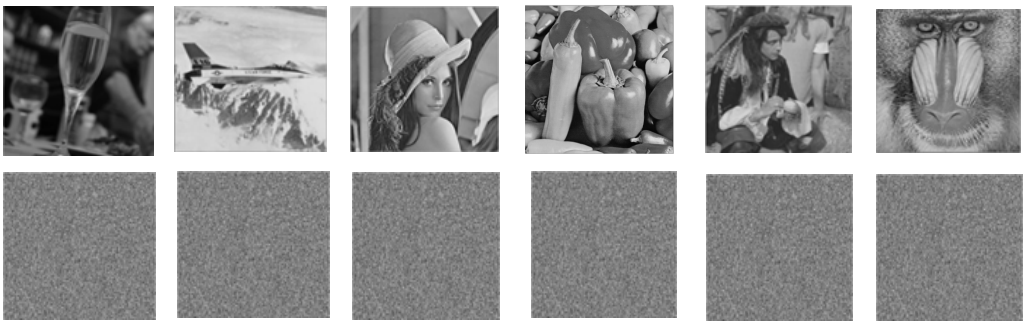
$$x'_{i,j} = \begin{cases} x''_{i,j} - 127 & x''_{i,j} < 0 \\ x''_{i,j} + 127 & x'_{i,j} > 0 \end{cases} \text{ when } \tau_{i,j} = 1 \tag{12}$$

The process of restoring the original image X according to the prediction error image X' is as shown in Eq. (13). Finally, the legal recipient gets the original image X.

$$x_{i,j} = \begin{cases} x'_{i,j}, & i = 1,\ j = 1 \\ x'_{i,j-1} + x'_{i,j}, & i = 1,\ j = 2,3 \dots, n \\ x'_{i,j} + x'_{i-1,j}, & i = 2,3 \dots, m,\ j = 1 \\ [\, x'_{i,j} + \frac{x'_{i-1,j-1} + x'_{i-1,j} + x'_{i,j-1}}{3}], & i = 2, \dots, m,\ j = 2, \dots, n, \end{cases} \tag{13}$$

3 Experiment results

In the following experiment, all the test images were gray-scale images of size 512×512. The experiment selects 6 grayscale images, which are (a)~(f), respectively. Encrypted 6 test images, the result is shown in Fig. 5.



(a) Cup (b) Airplane (c) Lena (d) Peppers (e) Man (f) Baboon

Figure 5: Six test images, the first action is the original test images, and the second action is the encrypted images

In Fig. 5, the first line is the original images, and the second line is the corresponding encrypted images. It can be seen that the generated encrypted image is random noise and the encrypted effect is good. In this section, we analyze the performance of the algorithm from two perspectives: maximum embedding capacity of the algorithm and security of the encrypted images.

3.1 Maximum embedding capacity

From Eq. (10), it can be seen that the maximum embedding capacity is related to the n_m and y_m. The optimal threshold y_m determines the optimal threshold range. First, we select 'Lena' and 'Baboon' images to test all thresholds and the maximum embedding rate. The results are shown in Tab. 2.

Table 2: All threshold ranges and corresponding embedding rates

y	Threshold range	Embedding rate under different threshold range (bpp)					
		Cup	Airplane	Lena	Peppers	Man	Baboon
1	[0, 1]	1.81	1.10	0.37	0.15	0.28	---
2	[-1, 2]	3.17	2.03	1.21	0.91	0.99	---
3	[-3, 4]	**3.92**	2.81	2.14	1.83	1.71	0.25
4	[-7, 8]	3.66	**2.88**	**2.77**	**2.66**	2.29	0.75
5	[-15, 16]	2.86	2.45	2.52	2.56	**2.30**	1.17
6	[-31, 32]	1.93	1.75	1.83	1.82	1.77	**1.19**
7	[-63, 64]	0.94	0.91	0.93	1.91	0.92	0.81

It can be seen from Tab. 2 that the maximum embedding rate of different images under different threshold ranges is different, and the maximum embedding rate under the optimal threshold is the highest. The higher the image texture is, the larger the optimal threshold range is. With the increase of the threshold range, the value of y_m also increases, this leads to the decrease of the maximum embedding rate of the texture image.

At the same time, the n_m and y_m values are determined by the prediction error image. Traditional prediction methods need to preserve some pixels when predicting. For example, the prediction method of the Xu's algorithm [Xu and Wang (2016)] needs to reserve 1/4 of the pixels to predict the remaining pixels, and the reserved pixels cannot be used to embed the information. The prediction method of the proposed algorithm only needs to reserve one pixel, and the remaining pixels are predicted to obtain the error. Although this may result in slightly lower prediction accuracy than the traditional prediction method, the number of pixels to be predicted is increased, the value of n_m can be increased. In order to analyze the maximum embedding rate of the algorithm in depth, we first define the prediction error ratio α:

$$\alpha = \frac{N}{m \times n} \tag{14}$$

where N is the number of prediction error values, and $m \times n$ is the size of an image. According to the prediction error ratio α, the embedding rate is redefined in combination with the Eqs. (10) and (11):

$$Rate = (8 - y_m) \times \alpha_p - \frac{(L_{\bar{\tau}} + L_\tau + L_a)}{M \times N} \approx (8 - y_m) \times \alpha_p \tag{15}$$

where α_p is the prediction error ratio of the proposed algorithm under the condition of $N = n_m$. The maximum predicts error ratio α_p of proposed algorithm is 1 because only

one pixel is reserved. However, the prediction error ratio of the algorithm of Xu or the traditional prediction method is $\alpha_{\mathrm{Xu}} \leq 0.75$ because they need to reserve 0.25 sample pixels of an image. Since a prediction error can only be embedded in one bit data, the maximum embedding rate of the Xu algorithm is:

$$Rate_{Xu} = \alpha_{Xu} \quad (\alpha_{Xu} \leq 0.75) \tag{16}$$

As can be seen from the proposed algorithm, the maximum value of y_m is 7, α_{p} value is greater than α_{Xu} in most cases. This proves that for most images, the maximum embedding rate of the proposed algorithm can be at least two times higher than that of Xu algorithm.

Then we tested the maximum embedding rate of the six common images in comparison with the literature [Yin, Luo and Hong (2014); Liu and Pun (2018); Xu and Wang (2016); Yi and Zhou (2017); Pauline and William (2018)]. The results are shown in Tab. 3. The 7 test images are Cup, Airplane, Lena, Peppers, Man and Baboon, which are all grayscale images of 512×512.

Table 3: Maximum embedding rate in different algorithms

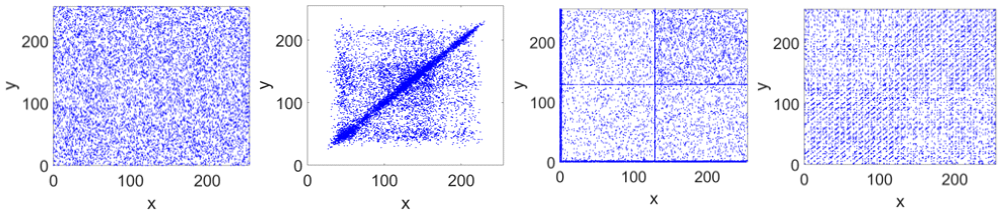
RDH-EI method	Maximum embedding rate (bpp)					
	Cup	Airplane	Lena	Peppers	Man	Baboon
Yin et al.	0.27	0.19	0.13	0.10	0.13	0.04
Xu et al.	0.58	0.36	0.30	0.25	0.27	0.11
Yi et al.	2.70	2.21	1.84	1.83	1.57	0.55
Liu et al.	1.99	1.69	1.58	1.61	1.39	0.59
Pauline et al.	0.99	0.99	0.99	0.99	0.99	0.99
Proposed	3.92	2.88	2.77	2.66	2.30	1.19

In Tab. 3, the algorithms [Yin, Luo and Hong (2014); Xu and Wang (2016)] uses the traditional reversible data hiding theory, in which [Yin, Luo and Hong(2014)] uses the histogram shifting method to embed information, and [Xu and Wang (2016)] combines the prediction error and histogram shifting method to embed information. Their algorithm can only embed 1 bit of information in one pixel, resulting in a low embedding rate. Literature [Yi and Zhou (2017)] and [Liu and Pun (2018)] are based on bit-plane block compression. It can be seen from Tab. 3 that the maximum embedding rate of the proposed algorithm is higher than [Yi and Zhou (2017); Liu and Pun (2018)]. This is because the proposed algorithm performs prediction errors and makes room for multiple bit planes. The embedding rate of the proposed algorithm is larger than [Pauline and William (2018)] since the proposed algorithm embeds information in multiple bit planes and [Pauline and William (2018)] only embeds information in one bit plane.

3.2 Security analysis

We focus on analyzing the security of Yin et al. [Yin, Luo and Hong (2014); Liu and Pun (2018); Xu and Wang (2016)] and the security of the proposed algorithm. A high security encrypted image should be able to resist such ciphertext-only attacks (COA) and known-

plaintext attacks. In this section, we first test the adjacent pixel correlation of the encrypted image. Then we will use ciphertext-only attack and known-plaintext attack to analyze the security of encrypted images.



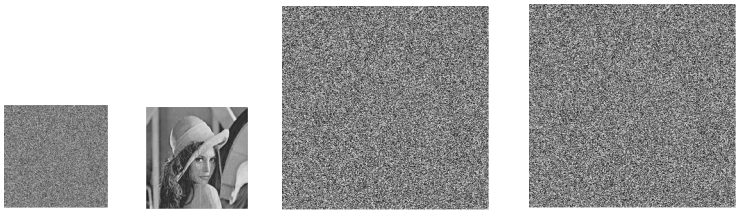
(a) Proposed (b) Yin (c) Xu (d) Liu

Figure 6: (a) Scatter plot of the encrypted image generated by the proposed algorithm, (b) Yin et al. [Yin, Luo and Hong (2014)] encrypted image scatter plot, (c) Xu et al. [Xu and Wang (2016)] encrypted image scatter plot, (d) Liu et al. [Liu and Pun (2018)] encrypted image scatter plot

The test plaintext image is 'Lena' grayscale image, and the plaintext image size is 512×512. We first encrypt the plaintext image using the proposed algorithm and the [Yin, Luo and Hong (2014); Liu and Pun (2018); Xu and Wang (2016)] algorithm. 10000 pixel values x and 10000 horizontal adjacent pixel values y in the encrypted image are randomly extracted. x and y form a coordinate point (x, y), and 10000 pairs (x, y) are displayed in a two-dimensional coordinate system in the form of a scatter plot. An encrypted image with better security should be able to completely disrupt the correlation of the original image, and its scatter plot should be randomly distributed. The proposed algorithm and [Yin, Luo and Hong (2014); Liu and Pun (2018); Xu and Wang (2016)] algorithm encryption image scatter diagram is shown in Fig. 6. As can be seen from Fig. 6, the pixel values of the encrypted image generated by the proposed algorithm are randomly distributed, and the encryption effect is better than the comparison algorithm.

3.2.1 Ciphertext-only attack

Ciphertext-only attack is an attack method in which an attacker only knows the ciphertext and cracks the encryption key according to the ciphertext image. Khelifi [Khelifi (2018)] proposed a ciphertext-only attack method for XOR encryption. The main goal of the attack method is to estimate the random matrix generated by secret key according to the ciphertext image. The ciphertext image is XORed by the estimated random matrix to obtain the plaintext content.



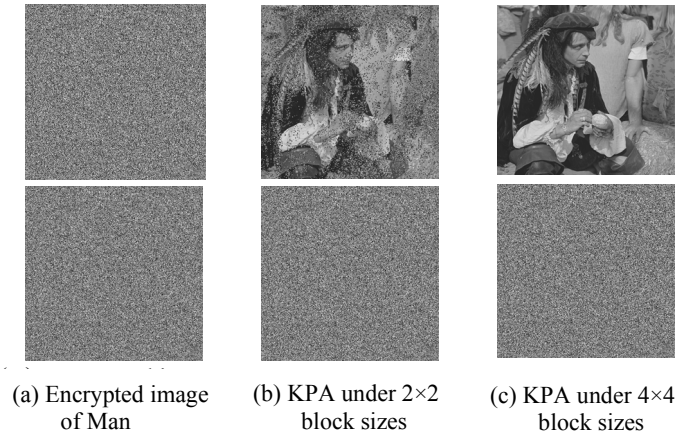
(a) 256×256 (b) 256×256 (c) 512×512 (d) 512×512

Figure 7: (a) The encrypted sampling pixels. (b) Decryption (a) with the estimated random matrix. (c) The encrypted image generated by the proposed algorithm. (d) The result of decrypting (c) with the estimated random matrix

Fig. 7(a) is the encrypted sampling pixels in the Xu algorithm, and (b) is the decryption (a) with the estimated random matrix. Although the original plaintext image is not completely decrypted by the Ciphertext-only attack method, the content of the plaintext image is already clear and discernible, so the Xu encrypted images can not resist the Ciphertext-only attack. Fig. 7(c) is the encrypted image generated by the proposed algorithm, (d) is the result of decrypting (c) with the estimated random matrix. Obviously, the decryption result is still random noise. The estimated random matrix has a correct rate of 50% per bit plane, and the attacker cannot obtain the correct key to decrypt the image. In conclusion, the proposed algorithm can resist Ciphertext-only attack.

3.2.2 Known-plaintext attack

Known plaintext attack assumes that the attacker can contact the encryption and decryption algorithm, and encrypt the known plaintext image to get the corresponding ciphertext image. The attacker uses plaintext and corresponding ciphertext to crack the encryption algorithm without knowing the encryption secret key. In 2016, Jolfaei et al. [Jolfaei, Wu and Muthukkumarasamy (2016)] proposed an attack method for permutation-only encrypted images. For 512×512 gray-scale images, attackers need at least three plaintext and corresponding ciphertext to achieve better attack results. Assuming that the attacker knows 'Lena' image and its corresponding encrypted image, the known plaintext attack algorithm proposed in Jolfaei et al. [Jolfaei, Wu and Muthukkumarasamy (2016)] is used to attack [Yin, Luo and Hong (2014); Liu and Pun (2018)] and the proposed algorithm respectively. The encrypted image of 'Man' is decrypted with the estimated scrambling matrix, and the result is shown in Fig. 8. In Fig. 8, the first line is the known plaintext attack result of the encrypted image generated by the [Yin, Luo and Hong (2014); Liu and Pun (2018)] algorithms at the block size of 2×2 and 4×4, and the second line is the known plaintext attack result of the encrypted image generated by the proposed algorithm.



(a) Encrypted image of Man (b) KPA under 2×2 block sizes (c) KPA under 4×4 block sizes

Figure 8: (a) Encrypted image of Man. (b) Known-plaintext attack under 2×2 block size (c) Known-plaintext attack under 4×4 block size. (The first line is the known plaintext attack result of the encrypted image generated by Yin et al. [Yin, Luo and Hong (2014)] and [Liu and Pun (2018)] algorithms, the second line is the known plaintext attack result of the encrypted image generated by the proposed algorithm.)

As can be seen from the results of Fig. 8, for the encrypted image of Yin et al. [Yin, Luo and Hong (2014); Liu and Pun (2018)], the attacker only needs to know one pair of plaintext and ciphertext to crack the content of the ciphertext image. The test results show that more than 50% of block scrambling coordinates can be recovered under 2×2 block size, and 100% of block scrambling coordinates can be recovered under 4×4 block size. It can be seen that block scrambling encryption schemes adopted by Yin et al. [Yin, Luo and Hong (2014); Liu and Pun (2018)] can not resist known plaintext attacks. The proposed algorithm adopts the combination of XOR and scrambling encryption to encrypt the original image. Under the known plaintext attack, the content of the encrypted image can not be leaked, therefore, the proposed algorithm can resist the known plaintext attack proposed in Jolfaei et al. [Jolfaei, Wu and Muthukkumarasamy (2016)].

4 Conclusion

This paper has presented a reversible data hiding algorithm for encrypted images based on prediction and adaptive classification scrambling to improve the embedding capacity and security performance of RHD-EI algorithm. The proposed algorithm uses a novel prediction error technique to compress the original image, and multiple bits of variable values can make space, so that one pixel can embed multi-bit information. Maximize the embedded capacity by adaptively selecting the optimal threshold. The proposed algorithm combines stream cipher with pixel scrambling encryption, improves the security of encrypted images. The embedded information can be extracted without loss and the decrypted image consistent with the original image can be retrieved. Experiment results have demonstrated that the proposed RDH-EI method can make more embedding space than the existing RDH-EI methods. At the same time, the encrypted image generated by the proposed algorithm has higher security performance than the comparative algorithm.

Funding Statement: This work has been supported by the National Natural Science Foundation of China (61872303, U1936113), the Science and Technology Innovation Talents Program of Sichuan Science and Technology Department (2018RZ0143) and the Key Project of Sichuan Science and Technology Innovation Pioneering Miaozi Project (19MZGC0163).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Cao, X.; Du, L.; Wei, X.; Dan, M.; Guo, X. (2016): High capacity reversible data hiding in encrypted images by patch-level sparse representation. *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1132-1143.

Chen, Y.; Yin, B.; He, H.; Yan, S.; Chen, F. (2018): Reversible data hiding in classification-scrambling encrypted-image based on iterative recovery. *Computers, Materials & Continua*, vol. 56, no. 2, pp. 299-312.

Das, R.; Baykara, M.; Tuna, G. (2019): A novel approach to steganography: Enhanced least significant bit substitution algorithm integrated with self-determining encryption feature. *Computer Systems Science and Engineering*, vol. 34, no. 1, pp. 23-32.

Fu, Y.; Kong, P.; Yao, H.; Tang, Z.; Qin, C. (2019): Effective reversible data hiding in encrypted image with adaptive encoding strategy. *Information Sciences*, vol. 494, pp. 21-36.

Jolfaei, A.; Wu, X.; Muthukkumarasamy, V. (2016): On the security of permutation-only image encryption schemes. *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 235-246.

Khelifi, F. (2018): On the security of a stream cipher in reversible data hiding schemes operating in the encrypted domain. *Signal Processing*, vol. 143, pp. 336-345.

Liu, Z.; Pun, C. (2018): Reversible data-hiding in encrypted images by redundant space transfer. *Information Sciences*, vol. 433-434, pp. 188-203.

Ma, K.; Zhang, W.; Zhao, X.; Yu, N. (2013): Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553-562.

Pauline, P.; William, P. (2018): An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images. *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1670-1681.

Qian, Z.; Zhang, X. (2016): Reversible data hiding in encrypted images with distributed source encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636-646.

Qin, C.; He, Z.; Luo, X.; Dong, J. (2018): Reversible data hiding in encrypted image with separable capability and high embedding capacity. *Information Sciences*, vol. 465, pp. 285-304.

Qin, C.; Zhang, W.; Cao, F.; Zhang, X.; Chang, C. (2018): Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection. *Signal Processing*, vol. 153, no. 1, pp. 109-122.

Qu, L.; He, H.; Chen, F. (2019): Reversible data hiding in encrypted image based on prediction error and classification scrambling. *Journal of Optoelectronics·Laser*, vol. 30, no. 2, pp. 168-174.

Shi, Y. Q.; Li, X.; Zhang X.; Wu, H. (2016): Reversible data hiding: advances in the past two decades. *IEEE Access*, vol. 4, no. 2, pp. 3210-3237.

Xu, D.; Wang, R. (2016): Separable and error-free reversible data hiding in encrypted images. *Signal Processing*, vol. 123, pp. 9-21.

Xiang, L. Y.; Wang, X. H.; Yang, C. F.; Liu, P. (2017): A novel linguistic steganography based on synonym run-length encoding. *IEICE transactions on Information and Systems*, vol. 100, no. 2, pp. 313-322.

Xiang, L. Y.; Wu, W. S.; Li, X.; Yang, C. F. (2018): A linguistic steganography based on word indexing compression and candidate selection. *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28969-28989.

Yi, S.; Zhou, Y. (2017): Binary-block embedding for reversible data hiding in encrypted images. *Signal Processing*, vol. 133, pp. 40-51.

Yin, Z.; Luo, B.; Hong, W. (2014): Separable and error-free reversible data hiding in encrypted image with high payload. *The Scientific World Journal*, vol. 2014, no. 1, pp. 604-876.

Yin, Z.; Abel, A.; Zhang, X.; Luo, B. (2016): Reversible data hiding in encrypted image based on block histogram shifting. *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2016, no. 1, pp. 2129-2133.

Zhou, J.; Sun, W.; Dong, L.; Liu, X. (2016): Secure reversible image data hiding over encrypted domain via key modulation. *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 26, no. 3, pp. 441-452.