High Capacity Reversible Data Hiding Scheme Based on AMBTC for Encrypted Images

Thai-Son Nguyen[1*], Chin-Chen Chang[2], Chia-Chen Lin[3]

[1] School of Engineering and Technology, Tra Vinh University, Vietnam
[2] Department of Information Engineering and Computer Science, Feng Chia University, Taiwan
[3] Department of Computer Science and Information Management, Providence University, Taiwan
thaison@tvu.edu.vn, ccc@cs.ccu.edu.tw, mhlin3@pu.edu.tw

Abstract

Reversible data hiding for encrypted images is a technique that allows for exact extraction of embedded data and precise reconstruction of the encrypted image to its original version. In this paper, we propose a novel, reversible data hiding technique based on an absolute moment block truncation coding (AMBTC) algorithm for encrypted images. In the proposed scheme, each selected block is represented by an AMBTC code to vacate space to embed data after image encryption. Upon obtaining the encrypted image with the embedded data, the receiver can separately extract the embedded secret data and also recover the cover image losslessly. The experimental results for the proposed scheme achieved excellent performance. The proposed scheme is significantly superior to state-of-the-art schemes in terms of embedding capacity and image quality.

Keywords: Reversible data hiding, AMBTC, Image encryption, Privacy protection, Reversibility

1 Introduction

Reversible data hiding (RDH) is a technique that aims to exactly extract embedded secret data and reconstruct the original image losslessly. Due to these properties, the RDH technique has attracted much interest from researchers in several fields, such as the military, for medical images, and law forensics. As such, many RDH schemes have been proposed in the literature and can be generally divided into three scheme types: a lossless-compression-based schemes [1-3], difference expansion (DE) based schemes [4-6, 10-11], and histogram shifting (HS) based schemes [7-9, 12].

Nowadays, the requirement to protect digital-data privacy in a cloud-computing environment is increasingly being sought out and the embedding of secret data into encrypted data may be useful in satisfying this requirement [14, 15, 30-34]. For example, a content owner can encrypt the original version of digital data before transmitting it to a data manager. The data manager then hides the secret data in encrypted images to facilitate management and authentication, even though the content of the original image is unknown to the data manager. Finally, when a receiver obtains the encrypted image with embedded data, the embedded secret data can be exactly extracted and the original version of the cover image completely reconstructed. In the last few years, many RDH schemes have been proposed for encrypted images (RDHEI). In [16], a well-known encryption algorithm, such as the advanced encryption standard (AES), was used to encrypt the original image. Then, for each encrypted block, one secret bit was embedded by using a LSB substitution technique. Conversely, an analysis of local standard deviation was used to obtain the directly decrypted image, and the embedded data. However, this scheme has a low embedding capacity (EC). In addition, the original version cannot be recovered losslessly from the encrypted version of the cover image. Subsequently, Zhang [17] divided the encrypted image into non-overlapping blocks. Then one secret bit was embedded into each encrypted block by simply flipping three LSBs of a predefined pixel set of this block. However, incorrect bits occurred in the extracted data after data extraction. Moreover, when a large amount of secret data is embedded, the cover image is permanently altered and cannot be reconstructed to its original version. Later on, Hong et al. [18] introduced an improved scheme thru the work in [17]. In this scheme, they used a different estimation function and side-match algorithm to exploit the spatial correlation of the pixels. By doing so, the error rate of the extracted bits was reduced significantly. The schemes in [17-18] use the spatial correlation of pixels to extract the embedded bits. This means that the receiver must decrypt the encrypted image to extract the embedded data. To separate the extraction of the secret data from the decryption of the cover image, Zhang [19] proposed a separable RDHEI scheme. Zhang's scheme generated free space that can carry the secret data by compressing LSB bits of the encrypted image. However, to recover the original image, extra information is required at the receiver side, and this scheme provided limited embedding capacity. To increase the free space to embed secret data, Zhang et al. [20] further improved the compression rate of the encrypted image by using low-density parity-check codes. To perfectly recover the original image, their scheme also required extra information. To achieve error-free extraction of the data, Yin et al. [21] proposed a RDHEI scheme based on multi-granularity encryption. In Yin et al.'s scheme, only smooth blocks were selected to embed the secret data. Definitely, in [17-21], preserved space was generated after image encryption (PRAE). As a result, the embedding space was limited due to the low redundancy in the encrypted images. To overcome this shortcoming, schemes based on preserving space before encryption (PRBE) were proposed in [22-23, 26].

*Corresponding Author: Thai-Son Nguyen; E-mail: thaison@tvu.edu.vn
DOI: 10.53106/160792642022032302006

To enhance the embedding capacity (EC) of the RDHEI schemes, Ma et al. [23] first designed an RDH scheme for encrypted images based on PRBE. Their scheme vacated the space by embedding LSB bits of a smooth region into a complex region by using a RDH scheme [13] before the image was encrypted. However, their scheme resulted in low image quality for the decrypted image since the PSNR is smaller than 45 dB for an embedding rate (ER) of 0.4 bpp. To further improve the performance of the previous work [23], Cao et al. [26] proposed a new RDH scheme for encrypted images that is based on patch-level sparse representation. In their scheme, a complete dictionary is trained by using the K-means singular value decomposition (K-SVD) [27-28]. Then, according to the generated dictionary, the sparse coding technique is used to construct a large vacated space for embedding data. Accordingly, Cao et al.'s scheme achieved better performance than Ma et al.'s scheme [23]. However, to maintain reversibility, the residual errors and the dictionary should be embedded into the encrypted images, resulting in the decrease of embedding capacity and image quality of the decrypted images. Therefore, the performance of Cao et al.'s scheme is unsatisfactory. Later on, in [33], Wang et al. proposed a new RDH scheme in encrypted images based on a block-based adaptive MSB encoding technique. In their scheme, the specific image encryption method preserves the relevance of the MSB bit planes in each block for data embedding. Wang et al.'s scheme has provided enhanced security, significantly improves the embedding rate, and obtains higher visual quality for the decrypted images. In 2018, Yin et al. [29] proposed a novel RDH scheme in encrypted AMBTC-compressed images. In their scheme, the quantization levels within an AMBTC compression code were encrypted by a XOR operation in advance to make sure the correlation between two quantization levels in an AMBTC compression code will be the same even after encryption. Finally, when two quantization levels are the same the bitmap replacement and the prediction error based RDH scheme were applied to vacate the room to embed the secret messages. The average embedding capacity provided by Yin et al.'s scheme is around 6081 bits.

Inspired by the advantages of state-of-the-art PRBE schemes, we propose a new RDHEI scheme based on PRBE concept. We observed that the redundancy of the pixels in the original version of the image was high. Therefore, if the pixels can be considerably compressed prior to image encryption and a large amount of space can be preserved to embed data in the encrypted images. According to this characteristic, the absolute moment block truncation coding (AMBTC) algorithm [24] was applied in the proposed scheme to preserve space prior to encrypt the image. For the content owner, each selected block is represented by AMBTC code and the corresponding compressed errors. By doing so, fewer bits are used to represent the block. As a result, large preserved space is obtained, meaning that more secret bits can be embedded. Upon obtaining the encrypted image with embedded data, the receiver can separately and exactly extract the embedded secret data, and recover the cover image losslessly. Experimental results demonstrated that the proposed scheme achieves excellent performance in three various aspects:

- Reversibility is achieved, meaning that in the proposed scheme, the cover image is reconstructed without any errors.
- Instead of only considering the high correlation of the pixels in each block to preserve space, we represent a block by using the AMBTC code, which improves upon the preserved space that can embed more data.
- The image quality of the directly decrypted image was significantly better than that of state-of-the-art schemes [18-23]. For instance, the PSNR of the proposed scheme was greater than 55 dB for an ER of 0.5 bpp, while the PSNR of our major competitor [23] is smaller than 40 dB.

The rest of the paper is organized as follows. Section 2 reviews the AMBTC algorithm. Section 3 describes the details of the proposed scheme, and our experimental results are presented and analyzed in Section 4. Finally, our conclusions are given in Section 5.

2 Related Work

2.1 Absolute Moment Block Truncation Coding

Absolute moment block truncation coding (AMBTC) [24] is a technique that is a variation of traditional block truncation coding (BTC). However, the AMBTC algorithm is computationally simpler than the BTC algorithm, while giving the same performance. In AMBTC, an absolute deviation is reserved with a mean value instead of by using standard deviation as in traditional BTC. Specifically, an image is first divided into non-overlapping blocks with a size of $n \times n$. Note that in AMBTC n is set to 4. Then, for each block i, the mean value $\overline{\eta}_i$ and the absolute deviation α_i are calculated as follows.

$$\overline{\eta}_i = \frac{\sum_{j=1}^{n \times n} x_{i,j}}{n \times n}, \tag{1}$$

$$\alpha_i = \frac{\sum_{j=1}^{n \times n} |x_{i,j} - \overline{\eta}_i|}{n \times n}, \tag{2}$$

where $x_{i,j}$ denotes the j^{th} pixel in the block i.

According to the mean value $\overline{\eta}_i$, a bitmap BM_i that consists of two different groups is determined as: If the value of pixels is smaller than $\overline{\eta}_i$, these pixels belong to group-0, and are denoted as 0 in the bitmap BM_i. Otherwise, these pixels are members of group-1, and are denoted as 1 in the bitmap BM_i. After encoding by AMBTC, the block i is represented by 31 bits of the AMBTC code $C_i = (\overline{\eta}_i, \alpha_i, BM_i)$, including the mean value $\overline{\eta}_i$ (8 bits), the absolute deviation α_i (7 bits), and the bitmap BM_i (16 bits).

In the decoding procedure, the low mean value L_i for group-0 and the high mean value H_i for group-1 can be computed by using Equations (3) and (4), respectively.

$$L_i = \overline{\eta}_i - \frac{n \times n \times \alpha_i}{2(n \times n - z)}, \tag{3}$$

$$H_i = \overline{\eta}_i + \frac{n \times n \times \alpha_i}{2z}, \tag{4}$$

Where z is the number of pixels having the value larger than or equal to $\overline{\eta}_i$. After two mean values, L_i and H_i, are obtained, the image block is reconstructed by replacing each

value 0 in BM_i by L_i and each value 1 in BM_i by H_i, respectively. Figure 1 shows an example of the AMBTC algorithm.

131	133	131	130
132	133	101	126
133	133	100	107
134	153	102	116

(a) Original block

1	1	1	1
1	1	0	1
1	1	0	0
0	1	0	0

(b) Bitmap *BM*

134	134	134	134
134	134	105	134
134	134	105	105
134	134	105	105

(c) Reconstructed block

Figure 1. Example of the AMBTC algorithm

Assume that the original image block i is shown in Figure 1(a). The mean value of this block is calculated as $\overline{\eta}_i = 125$, and the absolute deviation is determined as $\alpha_i = 12$. Then, each pixel is compared to the mean value $\overline{\eta}_i$ to generate the bitmap BM as shown in Figure 1(b). Finally, the block is represented by the AMBTC code as (125, 12, 1111110111001100), which is sent to a receiver. After obtaining the AMBTC code, the receiver can reconstruct the image block, shown in Figure 1(c), by the decoding process.

2.2 RDH Scheme in Encrypted Images by Patch-Level Spare Representation

To further improve the performance of Ma et al.'s scheme [23], Cao et al. [26] proposed a new RDH for encrypted images based on patch-level sparse representation. Figure 2 shows the main processes of their scheme. In this scheme, to embed the secret data into the encrypted image, the dictionary is first constructed by using K-SVD technique for reserving room to hide data. Note that, the K-SVD training is an off-line procedure, and the corresponding dictionary produced by training is then considered fixed for the whole reversible data hiding procedure. A given cover image is divided into patches, also called as blocks, and they are encoded by using sparse coding according to the dictionary. By using the dictionary, one patch can be represented as a sparse combination of several atoms in the dictionary. Thanks to the representation power of sparse coding, only a small number of coefficients require space for recording. Thus, a higher capacity room is available. Then, the smoother patches with the lower residual errors are selected for embedding data. To generate embedding space, the selected patches are indicated by sparse coefficients and residual errors. For reversibility reason, the residual errors and the dictionary are reversibly embedded into the non-selected patches by using RDH algorithm [13]. Obviously, by using the sparse coding, only a small number of coefficients is needed to record. As a result, the better embedding space is obtained in Cao et al.'s scheme. However, their scheme offered limited image quality because of using the non-selected patches for embedding the residual errors.

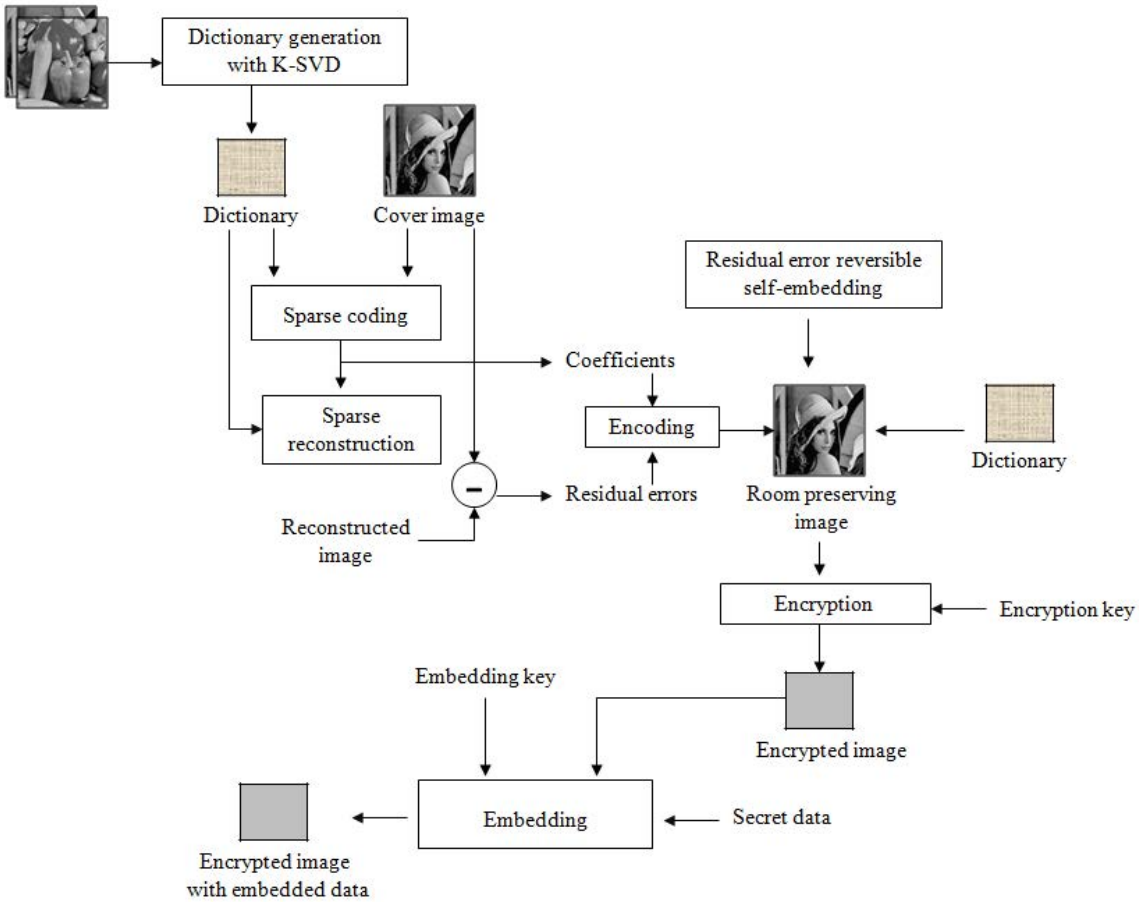


Figure 2. The main processes of Cao et al.'s scheme

We remark that, when the scheme based on PRBE mechanism can achieve the better performance by fully exploiting the spatial redundancy in the natural images for preserving the embedding space, it is more desirable to use fewer bits for representing the selected patches and to preserve the non-selected patches unchanged to design the RDH scheme based on PRBE for encrypted images.

3 Proposed Scheme

In this section, we propose a novel RDHEI scheme based on PRBE. The proposed scheme can be partitioned into three different phases: image encryption, embedding data in the encrypted image, and data extraction and image restoration. Figure 3 shows all of main steps of the proposed scheme.

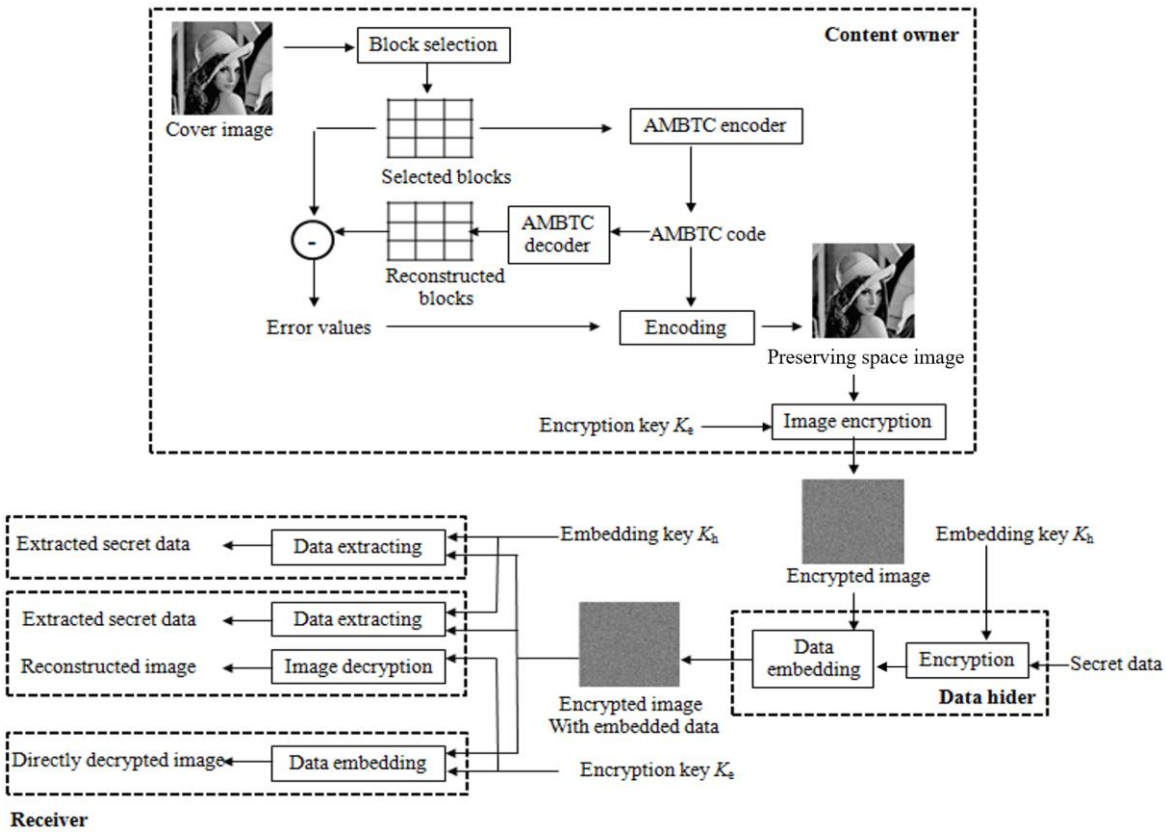


Figure 3. Framework of our proposed scheme

3.1 Image Encryption Phase

In this section, to encrypt cover image I, a content owner uses three sub-phases: block selection, self-reversible embedding, and image encryption. Assume that cover image I with a size of $H \times W$ is an 8-bit grayscale image, meaning that $I(a, b) \in [0, 255]$ and $1 \le a \le H$, $1 \le b \le W$. The content owner first divides the image I into pixel blocks that are then encoded by an AMBTC algorithm. After that, smoother blocks with lower compressed errors are determined and selected to preserve space to embed data. The selected blocks are encoded by AMBTC code and their corresponding compressed errors. Finally, the image with preserved space is encrypted to generate the encrypted image.

3.1.1 Block Selection

To achieve greater preserved space prior to encryption, in this sub-phase, smoother blocks with lower compressed errors is defined as suitable blocks should be determined and encoded by the AMBTC algorithm [24]. This process leads to the large free space that is generated in each block for carrying secret data. Given cover image I with a size of $H \times W$, we first divide it into non-overlapping image blocks with a size of $n \times n$, where n is set to 4 as default in the proposed scheme. Let M be the number of blocks P_i of the image I, and $= \frac{H \times W}{n \times n}$, so $I = \{P_i | \ i = 1, 2, \ldots M\}$.

Using the AMBTC algorithm, we encode each image block P_i into the AMBTC code where i is ranged from 1 to M, as $C_i = encode(P_i)$, and decode C_i to generate a recovered image block, as $Pc_i = decode(C_i)$, where $encode(.)$ and $decode(.)$ are AMBTC's encoding and decoding functions, respectively. Then, the corresponding compressed errors for E_i are calculated as:

$$E_i = P_i - Pc_i. \tag{5}$$

The image block P_i is considered as in a smooth region if Equation (6) holds.

$$-T \le \|E_i\| < T, \tag{6}$$

where ||.|| is the function that checks each value of E_i, and T is a threshold that is used to determine whether the block is a smooth block or a complex block. Distinctively, the block is determined as a smooth block by comparing its corresponding compressed errors E_i with threshold T. If all of values of E_i are smaller than T, the block belongs to the smooth region and is selected to embed data in the proposed scheme.

Along with the corresponding compressed errors E_i, the AMBTC code C_i is the representation information of image block P_i and it is used to reconstruct P_i as:

$$P_i = decode(C_i) + E_i, \tag{7}$$

Where $i = 1, 2, \ldots, M$. According to characteristics of the AMBTC algorithm, C_i is represented by $n_{Ci} = 31$ bits of a mean value $\overline{\eta}$, an absolute central deviation α, and a bitmap BM_i, as described in Section 2. For each value of the corresponding compressed errors E_i, we convert it into number $\lceil log_2(2 \times T) \rceil$ of bits. Therefore, the encoded bits of E_i are labeled as error bits and denoted as $n_{Ei} = n^2 \times \lceil log_2(2 \times T) \rceil$.

3.1.2 Self-reversible Embedding

Assume that several selected blocks P_l in the smooth region R_s, such as $R_s = \{P_l | l = 1, 2, \ldots, L\}$, are determined in the block selection sub-phase. Here, L is the number of selected smooth blocks that are determined by the size of the secret data S to be embedded. To preserve the vacated space to embed data, each image selected block P_l is indicated by the AMBTC code C_l and the corresponding error E_l. In addition, a position parameter that is used to point to the next selected block for data hider is also embedded into the block. This position parameter is represented by n_p bits, which is calculated as $n_p = \left\lceil log_2 \frac{H}{n} \right\rceil + \left\lceil log_2 \frac{W}{n} \right\rceil$. Therefore, for each selected block, the vacated room is preserved with $n_v = 8 \times n^2 - n_{Ci} - n_{Ei} - n_p$ bits to embed the secret data. It can be seen that, in the proposed scheme, for a cover image with a size of 512 ×512, $n = 4$, and $T = 2$, the embedding space of each selected block that is generated by the proposed scheme is 51 bits. This result outperforms the 48 bits obtained by the previous PRBE scheme [23]. In the proposed scheme, the number of selected smooth blocks, L is calculated by Equation (8).

$$L = \frac{|S|}{n_v} = \left\lceil \frac{|S|}{8 \times n^2 - n_{Ci} - n_{Ei} - n_p} \right\rceil$$
$$\text{and } L = \left\lceil \frac{|S|}{8 \times n^2 - 31 - n^2 \times \lceil log_2(2 \times T) \rceil - \left\lceil log_2 \frac{H}{n} \right\rceil + \left\lceil log_2 \frac{W}{n} \right\rceil} \right\rceil \tag{8}$$

Eventually, for reversibility, the selected smooth blocks P_l are represented by the AMBTC code C_l, followed by the corresponding compressed errors E_l. And we fill the preserved space with random bits. Therefore, the cover image I has been converted into an image with preserved space, I_r.

3.1.3 Image Encryption

When obtaining the image with preserved space, I_r, with the encryption key K_e, the content owner uses a stream cipher, i.e., RC4 or DES [25], to encrypt I_r and obtain the encrypted image E. Let $I_r(a, b)^k$ be the eight bits of the pixel $I_r(a, b)$, where $I_r(a, b)$ can be represented by:

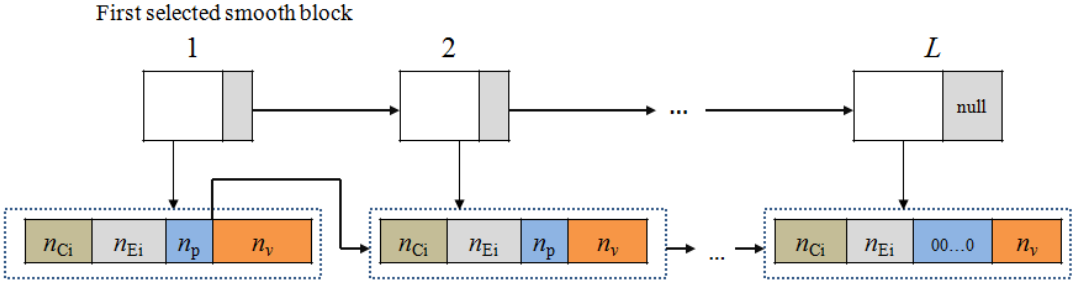
$$I_r(a, b)^k = \lfloor I_r(a, b)/2^k \rfloor mod\ 2, \quad k = 1, 2, \ldots 7 \tag{9}$$

where $\lfloor . \rfloor$ is a floor function. Then each encrypted bit $E_r(a, b)^k$ can be expressed by:

$$E_r(a, b)^k = I_r(a, b)^k \oplus r(a, b)^k, \quad k = 1, 2, \ldots 7 \tag{10}$$

where $r(a, b)^k$ is pseudo random bit that is generated by the stream cipher with the encryption key K_e.

After encryption of the pixels, we set n_p bits of position parameter in the selected smooth blocks to show the data hider where the next selected block is located. Figure 4 shows the structure for setting the n_p. Therefore, the data hider has the ability to access to the preserved space by following the structure.



Notations	Descriptions
n_{Ci}	n_{Ci} bits of AMBTC code C_i
n_{Ei}	The encoded bits of corresponding errors E_i
n_p	n_p bits of position parameter of the next selected block
n_v	The size of the preserved room

Figure 4. Structure for setting parameters

It is noted that the end indicator is set in the L^{th} selected block with n_p bits of "0". In addition, the position parameter of the first selected block and the size n_v of the preserved space for embedding data, both are embedded into the encrypted image by the RDH scheme proposed in [13]. Eventually, encrypted image E is obtained.

We remark that, without the encryption key K_e, it is very difficult for the data hider to access the original content of the cover image. Therefore, the security and the privacy of the cover image are preserved.

3.2 Embedding Data in An Encrypted Image

After obtaining the encrypted image E, the data hider can embed the secret data into encrypted image E to either manage or verify it, even though they are not able to access the original content of the cover image. The content owner has been hidden the position parameter of the first selected block and the size n_v of the preserved space into E; therefore, the data hider can extract this information from encrypted image E to determine where and how many free bits are available to embed the secret data. To embed the secret data, the data hider searches each selected smooth block in encrypted image E, and performs the process of bit substitution to modify the corresponding bits of the reserved space by the secret bits. In this phase, to completely extract the embedded secret data, secret data S and its size S_z are embedded into encrypted image E, where S_z is the fixed value associated with the specific requirement and it is embedded before embedding secret data S.

After embedding secret data S, the position parameter of the first selected smooth block and size n_v of the embedding space are also concealed into the encrypted image with embedded data, E', by using the RDH technique [13] in the proposed scheme. For security reason, the secret data S is encrypted by the stream cipher with data hiding key K_h. Therefore, anyone who does not possess this key cannot reveal the embedded secret data S.

3.3 Extraction of the Secret Data and Reconstruction of the Cover Image

Upon receiving the encrypted image with embedded data, E', and according to which keys, encryption and/or data hiding keys, i.e., K_e and K_h, the receiver possesses, the secret data can be extracted and the cover image reconstructed. Three different cases are considered as follows.

3.3.1 Case 1: Extracting Data with Only the Data Hiding Key

Assume that the receiver only has the data hiding key K_h to authenticate and update the information for the cover image. To protect the security and privacy of the image content, the receiver may only have the ability to access the embedded data in the encrypted domain with the data embedding key. First, the receiver can extract the position parameter of the first selected smooth block and the size n_v of embedding space from E'. Then, image E' is partitioned into non-overlapped $n \times n$ blocks. After that, the embedded secret data are extracted by reading the last n_v bits from the selected smooth blocks in E'. Notably, the first S_z bits that are extracted represent the size of the embedded secret data S. Finally, the original version of the secret data S is reconstructed with data hiding key K_h. It is obvious that the whole process is implemented completely in the encrypted domain, which prevents leakage of the original content of the cover image.

3.3.2 Case 2: Decrypting the Image with Only the Encryption Key

Assume that receiver only has the encryption key K_e in this case. To decrypt the encrypted image with embedded data E', the receiver also extracts the position parameter of the first selected smooth block by using the RDH scheme [13], then, all of the selected smooth blocks are determined completely. Afterwards, image E' is partitioned into non-overlapped blocks that are decrypted according to two different cases.

i*) For unselected blocks, the decrypted version can be directly obtained by:*

$$I_r(a,b)^k = E_r(a,b)^k \oplus r(a,b)^k, \quad k=1, 2, \ldots 7 \tag{11}$$

where $r(a,b)^k$ is a pseudo random bit generated by the stream cipher with the encryption key K_e, and $I_r(a,b)^k$ and $E_r(a,b)^k$ are the decrypted bit and encrypted bit of the pixel , respectively. Note that the unselected blocks are decrypted losslessly.

ii) For selected blocks, the encoded bits are decrypted by encryption key K_e. Then, the first n_{Ci} bits of AMBTC code C_i are read and decoded by AMBTC to generate the decrypted block P_i as following.

$$P_i = decode(C_i), \tag{12}$$

where *decode*(.) is AMBTC decoding function, and i = 1, 2, …, L. After unselected and selected blocks both are decrypted completely, the directly decrypted image is obtained. Obviously, the difference between the original image I and the directly decrypted image is perceptible because the difference values are equal to the corresponding compressed errors E_i that is limited by the value of threshold T. Therefore, the smaller of the value of threshold T is, the better the visual quality of the directly decrypted image.

3.3.3 Case 3: Extracting the Secret Data and Recovering the Cover Image with Data Hiding and Encryption Keys

After the receiver obtains the encrypted image with embedded data, if the receiver also possesses both data hiding K_h and encryption key K_e, the receiver can extract the secret data exactly from E' and reconstruct the original version of the image precisely if required. Herein, with the encryption key K_e, the receiver can first decrypt image E' to achieve the directly decrypted image. Moreover, with data hiding key K_h, the receiver can extract the secret data completely, and obtain the corresponding compressed error E_i. Then, the original version of the selected block P_i is reconstructed by Equation (13).

$$P_i = decode(C_i) + E_i, \quad \textbf{(13)}$$

Note that, once the pixels in the smooth region are reconstructed completely by AMBTC code C_i and the corresponding compressed error E_i, the original version of the cover image is recovered without any error.

3.4 An Example of Embedding Data in an Encrypted Image

To further clarify embedding data in the encrypted image, an example of the embedding process is provided. Assume that the threshold $T = 2$. In Figure 5(a), we assume that our original image block P has the size of 4 × 4. According to AMBTC algorithm, the block P_i is encoded to generate the AMBTC code C_i Figures 5(b). Then, we obtain the decoded block P_{Ci} and the error values E_i, as shown Figure 5(c) and Figure 5(d), respectively. Then each error value of E_i is converted by the number $\lceil log_2(2 \times T) \rceil$ of binary bits and concatenated into the AMBTC code C_i. For example, the error value of 1 or -1 is encoded by 01 or 11, where the first bit denotes the sign of the error value. The AMBTC code C_i and error values are encrypted by the RC4 [25] and concatenate the n_p bits of position parameter of the next selected block and the n_v random bits to generate the encrypted image E. Once, receiving the encrypted image E, the data hider performs the process of bit substitution to replace the n_v bits of the preserved space by the secret bits.

131	133	130	130
132	133	131	130
133	133	131	131
134	134	131	132

(a) Original block P_i

(b) AMBTC code C_i : (132, 1, 0100010011001110)

131	133	131	131
131	133	131	131
133	133	131	131
133	133	133	131

(c) Decoded block P_{Ci}

0	0	-1	-1
1	0	0	-1
0	0	0	0
1	1	0	1

(d) Error values E_i

Figure 5. Example of embedding data into encrypted image

4 Experimental Results

In this section, we report on the effectiveness and feasibility of the proposed scheme and several state-of-the-art schemes on six standard images as shown in Figure 6. The size of each image was 512 × 512 pixels. In this experiment, all the images are encrypted using the stream cipher RC4 [25]. We would like to compare the performance of our proposed scheme with that of state-of-the-art scheme [23] on Kodak images (http://ww.r0k.us/graphics/kodak/). The experiment was implemented on a PC with an Intel(R) Core™ i7-3770 CPU @ 3.4 GHz and 8 GB of RAM and Windows 7 Professional 64-bit operating system. All of the algorithms were performed by MATLAB 2014b.

(a) Lena (b) Airplane (c) Barbara

(d) Man (e) Peppers (f) Crowd

Figure 6. Six 512 × 512 grayscale test images

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

Figure 7. Results of the proposed scheme on image "Lena" for ER = 0.8bpp
(a) Original image; (b) Encrypted version; (c) Directly decrypted version with PSNR 56.89 dB; (d) Reconstructed version

In the proposed scheme, the peak signal-to-noise ratio (PSNR) was used to evalute the quality of the directly decrypted image. Figure 7 shows the results of the proposed scheme on the 'Lena' image for an embedding rate (ER) of 0.8 bpp. Here, the original and encrypted versions of the image are shown in Figures 7 (a)-(b). In this experiment, we set $T = 2$ and $n = 4$. If the receiver only possesses the data hiding key K_h, they can extract the embedded secret data losslessly. In addition, when only having the encryption key K_e, the receiver can achieve a directly decrypted image with high image quality (56.89 dB), as shown in Figure 7(c). Once the receiver possesses both encryption and data hiding keys, the original version of the image is reconstructed losslessly, as shown in Figure 7(d). In addition, to demonstrate the image encryption security, we calculate the information entropy of encrypted image. The entropy $H(V)$ is the most outstanding feature of the randomness that is calculated by Equation (14):

$$H(V) = -\sum_{i=0}^{N-1} P(v_i) \log_2 P(v_i), \quad \textbf{(14)}$$

where $V = \{v_0, v_1, \ldots, v_{N-1}\}$ is the source data and $P(v_i)$ is the probability of v_i. If the information entropy is close to the maximum value, this means that the encrypted image obtains the good properties of randomness. For a grayscale image, the pixel v_i is an integer in the range of [0, 255], therefore, each grayscale pixel should be encoded by 8 binary bits. As a result, the ideal value of information entropy is 8. In the proposed scheme, the average information entropy is appropriately 7.985. In particular, for the Figure 7 (b), the information entropy is 7.984, it means that the proposed scheme maintained the good security of the hidden data.

Table 1 provides a performance comparison of the proposed scheme of the directly decrypted image with different values of ER and T. In the proposed scheme, the distortion is determined by the difference between the directly decrypted images and its original version, which is equal to the compressed errors of Ei, since Ei is limited by the threshold T. Hence, when a larger T is used, then larger error values of Ei are obtained. Table 1 shows that a larger T leads to poorer quality in the directly decrypted images. In addition, when the larger value of T is used, the more bits are required to represent the larger error values of Ei, which also leads to shrinkage of embedding space. Therefore, only T = 2 or 3 are used for testing to maintain higher embedding space and good image quality. In this experiment, n = 4 was used, and the PSNR value was used to evaluate the performance of the proposed scheme. Table 1 also shows that when the value of ER increases, the PNSR gradually decreases. Here, we vary the embedding rate (ER) from 0.1 bpp to its maximum with the step size 0.1.

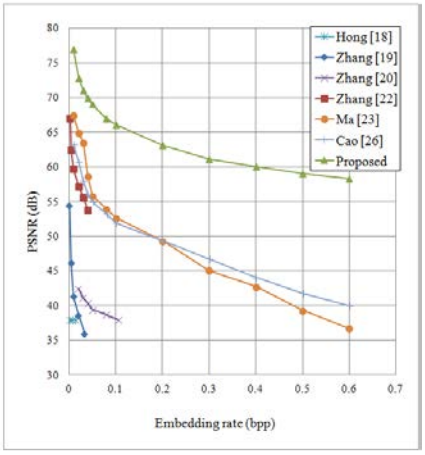
Table 1. Performance comparison of the directly decrypted image with different values ER and T ($n = 4$)

Images	ER	0.1	0.2	0.3	0.4	0.5	0.6	0.7	Average
Lena	$T = 2$	66.09	63.09	61.12	60.01	59.03	58.25	57.56	60.74
	$T = 3$	62.65	59.48	57.20	56.03	54.98	54.11	53.50	56.85
Barbara	$T = 2$	63.42	60.11	57.89	56.47	55.46	54.60	53.28	57.32
	$T = 3$	60.03	58.57	55.79	54.53	53.64	52.43	51.67	55.23
Airplane	$T = 2$	64.15	61.29	59.75	58.87	58.30	57.93	57.44	59.68
	$T = 3$	60.62	57.81	56.05	54.97	54.24	53.63	53.30	55.80

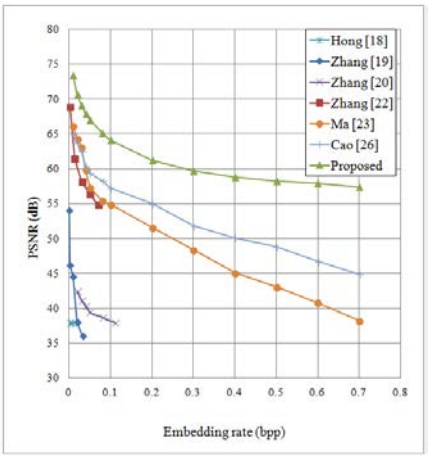
To demonstrate the high performance of the proposed scheme, we compared it with state-of-the-art schemes [18-20, 22-23, 26] in terms of image quality of directly decrypted image and embedding capacity. Six state-of-the-art schemes were used in this experiment because they achieved high performance by using the RDHEI technique. The reason we did not compare our proposed scheme with Yin et al.'s scheme [29] because their average embedding capacity is around 6081 bits.

As mentioned above, scheme [18] may give some errors when the secret data is extracted. Whereas, the proposed scheme and the other five schemes [19-20, 22-23, 26] are free

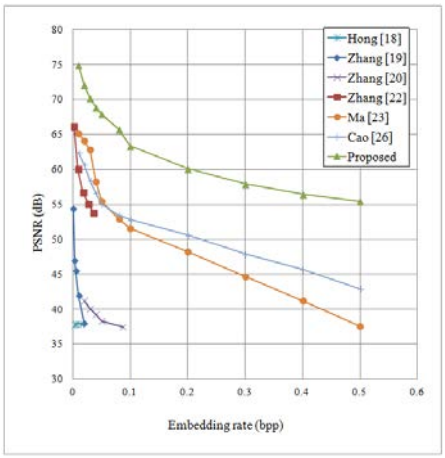
of any errors for various types of cover images. Figure 8 shows comparison results of the proposed scheme with six previous schemes [18-20, 22-23, 26] on the six test images. As shown in Figure 8, the image quality of the directly decrypted images is listed for various ER in the curve behaviors. From Figure 8, it is obvious that the top curves of different images are the performance results of the proposed scheme. These results mean that for a given ER, the image quality of the directly decrypted images obtained by the proposed scheme is significantly higher in the ER range than those of six state-of-the-art schemes [18-20, 22-23, 26]. Specifically, for six test images as shown Figure 6, when the ER was 0.4 bpp, the PSNRs of the proposed scheme were 60.01 dB, 58.82 dB, 56.47 dB, 59.53 dB, 58.12 dB, and 61.33 dB, while the results of the closest competitor [26] were 44.06 dB, 50.13 dB, 45.71 dB, 47.23 dB, 43.77 dB, and 52.24 dB, respectively. Here, two parameters K and L of the scheme [26] were set to 64 and 2 since this setting provided the best performance in their scheme. The proposed scheme obtained better image quality than the scheme [26]. This is because the scheme [26] is based on sparse coding, meaning that the selected blocks for preserving embedding space is decomposed into the sparse coefficient x and the residual errors e. Then, the coefficient x is represented by the index p and L non-zero coefficients that are reversibly embedded into the selected blocks. Therefore, the larger embedding space, 55 bits per the selected block, is obtained in their scheme, compared to only 51 per the selected block in the proposed scheme. However, for image lossless recovery, the scheme [26] also used the non-selected block to reversibly embed the residual errors by using RDH scheme [13]. By doing so, the directly decrypted image is distorted significantly. In contrast, even the compressed errors were also generated in the proposed scheme, and they were embedded into the selected blocks, instead of embedding into the non-selected blocks as was done in [26]. This process leads to the decrease of embedding space in the selected blocks of the proposed scheme. However, the non-selected blocks are remained unchanged during our embedding process, resulting in the image quality of these non-selected blocks are preserved. Accordingly, the high image quality of the directly decrypted images is achieved by our proposed scheme. The high performance of the proposed scheme implied very good potential for real-time applications.



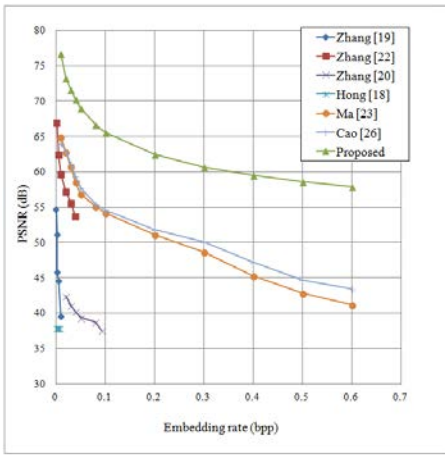
(a) Lena



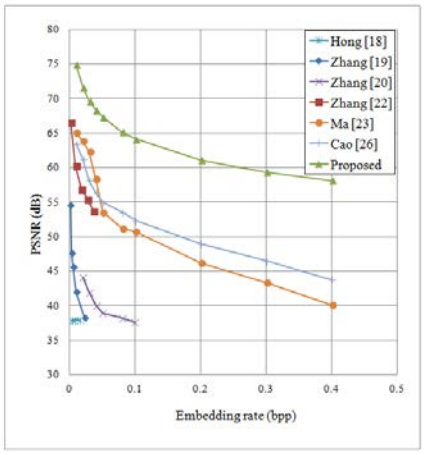
(b) Airplane



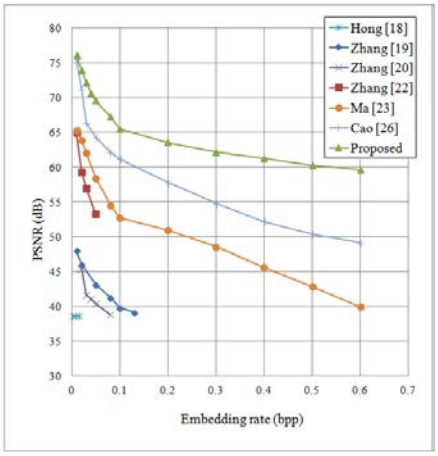
(c) Barbara



(d) Man



(e) Peppers



(f) Crowd

Figure 8. Comparisons of image quality for directly decrypted images with six state-of-the-art schemes [18-20, 22-23, 26]

Table 2. Comparison of PSNRs of directly decrypted images for Kodak images

ER	kodim01			kodim02			kodim03			kodim04			kodim05		
(bpp)	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed
0.05	53.29	54.31	71.13	55.21	55.86	69.40	57.38	61.14	71.45	55.12	54.89	70.15	53.19	54.82	70.16
0.10	50.10	51.03	68.39	53.22	54.71	66.50	55.39	59.22	69.20	53.27	53.20	67.46	50.67	49.50	67.39
0.15	47.68	49.61	66.57	51.64	53.78	64.91	53.27	57.59	67.67	51.64	51.58	65.49	48.23	48.04	65.76
0.20	45.44	47.92	65.44	50.54	52.77	63.73	51.98	55.76	66.52	50.23	50.52	63.97	45.72	46.65	63.82
0.25	43.29	46.13	64.54	49.19	51.75	62.83	50.46	54.81	65.56	48.61	49.50	62.85	43.91	44.21	61.45
0.30	41.08	44.06	63.28	47.87	50.72	62.14	48.84	53.75	64.76	46.90	48.57	62.06	42.49	42.73	59.64
Average	46.81	48.84	66.56	51.28	53.27	64.92	52.89	57.05	67.53	50.96	51.38	65.33	47.37	47.66	64.70
ER	kodim06			kodim07			kodim08			kodim09			kodim10		
(bpp)	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed
0.05	54.02	62.78	73.37	56.78	61.02	71.59	53.21	52.16	71.50	56.61	57.03	68.92	56.43	56.81	69.94
0.10	52.28	59.60	70.02	54.86	57.25	68.54	49.64	48.58	67.71	54.38	55.98	66.14	54.01	54.59	66.82
0.15	50.97	54.83	68.40	53.36	55.64	66.78	46.98	45.34	64.26	53.28	54.22	64.44	52.29	53.16	64.92
0.20	49.58	51.32	67.27	51.89	53.80	65.56	43.27	43.22	61.81	50.97	52.85	63.11	50.67	51.93	63.60
0.25	46.88	50.97	66.19	50.23	52.34	64.61	41.38	40.84	59.02	49.32	51.84	62.11	49.09	50.98	62.62
0.30	44.39	50.81	65.44	48.71	52.06	63.87	39.71	38.76	56.27	47.86	50.49	61.30	47.48	49.92	61.84
Average	49.69	55.05	68.45	52.64	55.35	66.83	45.70	44.82	63.43	52.07	53.74	64.34	51.66	52.90	64.96
ER	kodim11			kodim12			kodim13			kodim14			kodim15		
(bpp)	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed
0.05	55.06	61.48	70.96	55.02	57.84	70.61	51.09	50.88	72.04	53.67	53.11	70.23	55.24	59.05	70.81
0.10	53.37	59.20	69.05	53.46	56.18	68.19	47.53	47.22	69.55	51.04	49.71	66.97	54.03	57.62	68.00
0.15	51.62	56.62	67.65	52.69	54.69	66.74	44.34	43.79	66.57	48.89	48.64	65.58	52.49	55.89	66.24
0.20	50.10	54.55	66.63	51.22	52.89	65.71	40.85	40.30	63.43	46.61	46.57	63.16	50.94	54.72	65.13
0.25	47.94	53.19	65.82	49.64	52.11	64.61	38.41	38.02	61.04	44.84	44.93	61.46	49.43	53.87	64.26
0.30	45.27	51.58	65.17	47.99	51.19	63.62	35.74	35.96	28.92	43.21	43.75	59.34	48.07	52.72	63.52
Average	50.56	56.10	67.55	51.67	54.15	66.58	42.99	42.70	60.26	48.04	47.79	64.46	51.70	55.65	66.33
ER	kodim16			kodim17			kodim18			kodim19			kodim20		
(bpp)	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed
0.05	57.67	60.49	73.03	55.21	56.33	69.98	54.62	54.21	68.18	56.51	55.83	68.60	-	64.18	85.60
0.10	55.52	58.13	69.47	53.82	54.62	67.23	51.01	50.01	65.14	54.33	53.65	65.79	-	62.77	85.54
0.15	53.69	56.64	67.66	52.41	53.14	65.42	48.41	48.71	62.65	52.64	51.76	64.03	-	61.67	85.30
0.20	51.34	54.74	66.31	50.93	51.34	64.15	45.98	47.14	60.19	50.45	49.81	62.79	-	60.36	84.82
0.25	49.37	53.21	65.20	48.36	50.49	63.15	44.49	45.69	57.92	48.39	48.92	61.89	-	60.08	84.52
0.30	47.86	51.37	64.31	47.71	49.55	62.42	42.94	44.15	55.41	46.04	48.12	61.05	-	59.71	84.33
Average	52.58	55.76	67.66	51.41	52.58	65.39	47.91	48.32	61.58	51.39	51.35	64.03	0.00	61.46	85.02
ER	kodim21			kodim22			kodim23			kodim24					
(bpp)	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed	Ma et al.	Cao et al.	Proposed			
0.05	56.84	58.74	70.21	52.09	53.76	69.01	56.81	58.89	71.10	54.21	57.84	89.34			
0.10	54.61	56.39	67.38	52.25	52.61	66.23	55.13	57.68	68.12	52.55	56.07	80.38			
0.15	52.82	54.58	65.57	50.82	51.73	64.44	53.34	55.78	66.47	50.79	54.49	73.79			
0.20	50.64	52.87	64.21	49.43	50.52	63.24	52.01	53.91	65.20	49.26	52.84	71.17			
0.25	49.09	51.62	63.22	47.86	49.51	62.27	50.36	52.96	64.25	46.07	51.44	68.43			
0.30	46.92	50.30	62.37	45.33	48.37	61.53	48.72	51.99	63.42	43.97	49.93	66.16			
Average	51.82	54.08	65.49	49.63	51.08	64.45	52.73	55.20	66.43	49.48	53.77	74.88			

Table 2 shows comparison results for the proposed scheme and two previous PRBE schemes [23, 26]. The results of the proposed scheme were considerably better than those of Ma et al.'s scheme [23] and Cao et al.'s scheme [26] in most cases. The gains of average PSNR from the proposed scheme for Kodak images were from 12.27 dB to 25.40 dB and from 10.48 dB to 23.56 dB, respectively. As such, it could be concluded that the proposed scheme was more effective for RDHEI.

5 Conclusion

In this paper, we proposed a new EIRDH scheme based on an AMBTC mechanism. The proposed scheme inherits the advantage of preserving space prior to image encryption. By representing the information of each selected image block into the AMBTC code, the proposed scheme achieves a substantial amount of preserved space to embed data. Subsequently, to embed the secret data, the data hider simply uses bit substitution in a data embedding phase. The experimental results indicated that the proposed scheme achieves excellent performance in reversibility, and a higher PSNR, while maintaining a large EC. Specifically, compared with image quality results from current state-of-the-art schemes, the gains from average PSNRs of the proposed scheme are always larger than 10 dB, which implies that the proposed scheme is very suitable for real-time applications.

Acknowledgments

This study was fully supported by the Tra Vinh University under grant contract number 207/HĐ.HĐKH - ĐHTV.

References

[1] J. Fridrich, M. Goljan, R. Du, Lossless Data Embedding for All Image Formats, *SPIE proceedings of Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents*, Vol. 4675, San Jose, CA, USA, 2002, pp. 572-583.

[2] M. Celik, G. Sharma, A. Tekalp, E. Saber, Lossless Generalized-LSB Data Embedding, *IEEE Transactions on Image Processing*, Vol. 14, No. 2, pp. 253-266, February, 2005.

[3] C. C. Chang, T. S. Nguyen, A Reversible Data Hiding Scheme for SMVQ Indices, *Informatica*, Vol. 25, No. 4, pp. 523-540, September, 2014.

[4] 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, August, 2003.

[5] Y. Hu, H. K. Lee, J. Li, DE-based Reversible Data Hiding with Improved Overflow Location Map, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 2, pp. 250-260, February, 2009.

[6] D. M. Thodi, J. J. Rodriguez, Expansion Embedding Techniques for Reversible Watermarking, *IEEE Transactions on Image Processing*, Vol. 16, No. 3, pp. 721-730, March, 2007.

[7] T. S. Nguyen, C. C. Chang, T. H. Shih, Effective Reversible Image Steganography Based on Rhombus Prediction And Local Complexity, *Multimedia Tools and Applications*, Vol. 77, No. 20, pp. 26449-26467, October, 2018.

[8] P. H. Vo, T. S. Nguyen, V. T. Huynh, T. N. Do, A Novel Reversible Data Hiding Scheme with Two-Dimensional Histogram Shifting Mechanism, *Multimedia Tools and Applications*, Vol. 77, No. 21, pp. 28777-28797, November, 2018.

[9] C. C. Chang, T. S. Nguyen, C. C. Lin, Reversible Image Hiding for High Image Quality Based on Histogram Shifting and Local Complexity, *International Journal of Network and Security*, Vol. 16, No. 3, pp. 208-220, May, 2014.

[10] J. Li, X. Li, B. Yang, Reversible Data Hiding Scheme for Color Image Based on Prediction-Error Expansion And Cross-Channel Correlation, *Signal Processing*, Vol. 93, No. 9, pp. 2748-2758, September, 2013.

[11] X. Li, J. Li, B. Li, B. Yang, High-fidelity Reversible Data Hiding Scheme Based on Pixel-Value-Ordering and Prediction-Error Expansion, *Signal Processing*, Vol. 93, No. 1, pp. 198-205, January, 2013.

[12] T. S. Nguyen, P. H. Vo, Reversible Image Authentication Scheme Based On Prediction Error Expansion, *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 21, No. 1, pp. 253-262, January, 2021.

[13] L. X. Luo, Z. Y. Chen, M. Chen, X. Zeng, Z. Xiong, Reversible Image Watermarking Using Interpolation Technique, *IEEE Transactions on Information Forensics and Security*, Vol. 5, No. 1, pp. 187-193, March, 2010.

[14] Y. Wang, K. N. Plataniotis, An Analysis of Random Projection for Changeable and Privacy-Preserving Biometric Verification, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 40, No. 5, pp. 1280-1293, October, 2010.

[15] A. Dabrowski, E. Weippl, I. Echizen, Framework Based on Privacy Policy Hiding for Preventing Unauthorized Face Image Processing, *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Manchester, UK, 2013, pp. 455-461.

[16] W. Puech, M. Chaumont, O. Strauss, A Reversible Data Hiding Method for Encrypted Images, *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X. Proceedings of SPIE* 6819, San Jose, CA, USA, 2008.

[17] X. Zhang, Reversible Data Hiding in Encrypted Image, *IEEE Signal Processing Letters*, Vol. 18, No. 4, pp. 255-258, April, 2011.

[18] W. Hong, T. S. Chen, H. Y. Wu, An Improved Reversible Data Hiding in Encrypted Images Using Side Match, *IEEE Signal Processing Letters*, Vol. 19, No. 4, pp. 199-202, April, 2012.

[19] X. Zhang, Separable Reversible Data Hiding in Encrypted Image, *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 2, pp. 826-832, April, 2012.

[20] X. Zhang, Z. Qian, G. Feng, Y. Ren, Efficient Reversible Data Hiding in Encrypted Images, *Journal of Visual Communication and Image Representation*, Vol. 25, No. 2, pp. 322-328, February, 2014.

[21] Z. Yin, B. Luo, W. Hong, Separable and Error-Free Reversible Data Hiding in Encrypted Image with High

Payload, *The Scientific World Journal*, Vol. 2014, pp. 1-8, April, 2014.
[22] W. Zhang, K. Ma, N. Yu, Reversibility improved data hiding in encrypted images, *Signal Processing*, Vol. 94, pp. 118-127, January, 2014.
[23] K. Ma, W. Zhang, X. Zhao, N. Yu, F. Li, 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, March, 2013.
[24] M. Lema, O. R. Mitchell, Absolute Moment Block Truncation Coding And Its Application to Color Images, *IEEE Transactions on Communications*, Vol. 32, No. 10, pp. 1148-1157, October, 1984.
[25] W. Stallings, Cryptography and Network Security: Principles and Practice, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2003.
[26] X. Cao, L. Du, X. Wei, D. Meng, X. Guo, High Capacity Reversible Data Hiding in Encrypted Images by Patch-Level Sparse Representation, *IEEE Transactions on Cybernetics*, Vol. 46, No. 5, pp. 1132-1143, May, 2016.
[27] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, *IEEE Transactions on Signal Processing*, Vol. 54, No. 11, pp. 4311-4322, November, 2006.
[28] R. Rubinstein, T. Peleg, M. Elad, Analysis K-SVD: A Dictionary-Learning Algorithm for the Analysis Sparse Model, *IEEE Transactions on Signal Processing*, Vol. 61, No. 3, pp. 661-677, February, 2013.
[29] Z. Yin, X. Niu, X. Zhang, J. Tang, B. Luo, Reversible Data Hiding in Encrypted AMBTC images, *Multimedia Tools and Applications*, Vol. 77, No. 14, pp. 18067-18083, July, 2018.
[30] T. S. Nguyen, C. C. Chang, W. C. Chang, High Capacity Reversible Data Hiding Scheme for Encrypted Images, *Signal Processing: Image Communication*, Vol. 44, pp. 84-91, May, 2016.
[31] T. S. Nguyen, A Novel Reversible Data-Hiding Method Using Adaptive Rhombus Prediction and Pixel Selection, *International Journal of Network Security*, Vol. 23, No. 4, pp. 725-733, July, 2021.
[32] I. C. Dragoi, D. Coltuc, On the Security of Reversible Data Hiding in Encrypted Images by MSB Prediction, *IEEE Transactions on Information Forensics and Security*, Vol. 16, pp. 187-189, January 2021.
[33] X. Wang, C. C. Chang, C. C. Lin, Reversible Data Hiding in Encrypted Images with Block-Based Adaptive MSB Encoding, *Information Sciences*, Vol. 567, pp. 375-394, August, 2021.
[34] Y. J. Liu, C. C. Chang, T. S. Nguyen, High Capacity Turtle Shell-Based Data Hiding, *IET Image Processing,* Vol. 10, No. 2, pp. 130-137, February, 2016.

Biographies

Thai-Son Nguyen received the bachelor's degree in information technology from the Open University, HCM city, Vietnam, in 2005. From December 2006, he has been lecturer of Tra Vinh University, Tra Vinh, Vietnam. In 2011 and 2015, he received M.S. and PhD. degrees in computer sciences from Feng Chia University, Taichung, Taiwan. His current research interests include data hiding, image and signal processing, multimedia security, information security.

Chin-Chen Chang received the Ph.D degree in computer engineering from National Chiao Tung University, Hsinchu, in 1982. From July 1998 to June 2000, he was Director of the Advisory Office, Ministry of Education, R.O.C. From 2002 to 2005, he was a Chair Professor at National Chung Cheng University. From February 2005, he has been a Chair Professor at Feng Chia Univeristy. In addition, he was severed as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression, and data structures.

Chia-Chen Lin received the M.S. degree in information management and the Ph.D. degree in information management from Chiao Tung University, Hsinchu, Taiwan, in 1994 and 1998, respectively. She was a Visiting Associate Professor at Business School, University Illinois at Urbana Champain, during August 2006 to July 2007. She is currently a Professor in the Department of Computer and Information Management, Providence University, Sha-Lu, Taiwan. Her research interests include image and signal processing, image data hiding, mobile agent, and electronic commerce.