IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 3, 2019, accepted December 30, 2019, date of publication January 13, 2020, date of current version February 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966234

A High Capacity Reversible Data Hiding in Encrypted AMBTC-Compressed Images

GUO-DONG SU[1,2,3], (Member, IEEE), CHIN-CHEN CHANG[2,5], (Fellow, IEEE), AND CHIA-CHEN LIN[4], (Member, IEEE)

[1]School of Electronic and Information Engineering, Fuqing Branch of Fujian Normal University, Fuzhou 350300, China
[2]Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[3]Engineering Research Center for ICH Digitalization and Multi-Source Information Fusion (Fuqing Branch of Fujian Normal University), Fujian Province University, Fuzhou 350300, China
[4]Department of Computer Science and Information Management, Providence University, Taichung 433, Taiwan
[5]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Chia-Chen Lin (mhlin3@pu.edu.tw)

This work was supported in part by the Natural Science Foundation of Fujian Province of China under Grant 2018J01788, and in part by the Project of Ministry of Science and Technology of Taiwan under Grant MOS 108-2410-H-126-021.

ABSTRACT Recently, reversible data hiding in encrypted compressed images (RDHECI) has attracted more attention due to privacy information protection concerns. Meanwhile, AMBTC as one technique of lossy image compression that has lower storage costs and the simplicity in computation and is extensively used in many applications. Hence, this paper proposes an RDHECI scheme based on AMBTC, named O-AMBTC, to address privacy concerns. First, the original image is scrambled in a block-wise manner to generate the scrambled image which then is compressed by an AMBTC compression technique. Subsequently, the derived AMBTC compression codes are encrypted by using the methods of value modulation and stream cipher while the correlations between two quantization levels of AMBTC compression codes are retained and exploited to vacate redundant room to embed secret messages. Data hiding is then performed with the use of PBTL labeling strategy. In addition, another modified AMBTC compression code based RDHECI scheme, called M-AMBTC, is suggested to increase the ability to carry secret messages. In our dual approach, both the AMBTC-compressed image and the secret messages can be correctly recovered. Experimental results show that two proposed schemes are able to achieve average embedding rates as large as 0.6 bpp and 0.8 bpp when the block size is set to 2×2, respectively.

INDEX TERMS RDHECI, privacy protection, O-AMBTC, M-AMBTC, embedding rate.

I. INTRODUCTION

Image encryption is one of various techniques that are used to protect the privacy and security of image content and to prevent illegal access by an unauthorized third party. The modern cryptography [1], [2] methods ensure that an authorized third party can decrypt an image in a correct manner, while unauthorized parties cannot obtain any meaningful information. At the same time, academics have paid more attention to steganography [3], [4] for making private information secure and imperceptible to unauthorized persons. Steganography is a technique for data hiding. Reversible data hiding (RDH) is one branch of steganography, which is designed to embed secret data by slightly modifying the pixel values of the cover image while introducing a few distortions to the cover image and result in a stego-image [3], [4]. Notably, the key focus of RDH is that the original cover image can be recovered losslessly after the extraction of the hidden secret messages of the stego-image. In other words, an RDH-based scheme [5]–[20], [22]–[27] is a technique where the original image can be reconstructed completely at the recipient side. Therefore, it is used extensively for the secure transmission of military images or medical images.

Existing reversible data hiding based schemes can be roughly classified into the following four fundamental strategies: lossless compression based schemes [5], [6], difference expansion based schemes [7], [8], prediction error expansion based schemes [9], [10], and histogram shifting based schemes [11], [12]. A lossless compression based scheme usually conceals the secret messages into redundant space

The associate editor coordinating the review of this manuscript and approving it for publication was Kaitai Liang.

that is vacated using lossless compression techniques, e.g., Run Length Encoding and Huffman Coding. Schemes that are based on difference expansion usually embed the secret message into the least bit of difference value of an available pixel pair. Prediction error expansion based schemes produce prediction error values using a designed predictor, and then the expansions of these prediction error values are performed and employed to carry the secret messages. Histogram shifting based schemes usually conceal the secret message into a pixel value or prediction error value, whichever has the highest frequency in its histogram distributions. In summary, all of these schemes have the reversible property and maintain high quality in the stego-image.

In these various approaches, however, the content of the cover image is always exposed to the data hider during the embedding process. As a result, many RDH schemes in the encrypted domain have been proposed in an attempt to solve this problem. Reversible data hiding in encrypted images (RDHEI) is a technique that first encrypts the cover image and then transmits it to the data hider, such that the data hider conceals the secret messages into the encrypted image without knowledge of the content of the original image. Moreover, on the receiving side, the original cover image can be recovered losslessly and secret messages can be extracted separately. To the best of our knowledge, most existing RDHEI-based schemes are mainly designed to work in the un-compressed domain [13]–[18]. In [13], Zhang proposed a novel RDHEI scheme that first encrypted a cover image using the bit-wise based exclusive or (XOR) operation. Block division for encrypted image was then performed, and pixels in each block were segmented into two equal groups, S_0 and S_1. The 3 LSBs of pixels in S_0 were flipped if the to-be-hidden secret message in this block is '0'; and if the to-be-hidden secret message is '1', the 3 LSBs of pixels in S_1 were flipped. At the receiving side, two directly decrypted versions of one block, H_0 and H_1, can be derived by flipping all 3 LSBs of pixels in S_0 and S_1, respectively. A fluctuation function was designed to evaluate the degree of block's smoothness. The decrypted block with a smoother degree was judged as the original block. Also, the embedded message '0' can be extracted if the original block is judged as H_0; otherwise, the embedded message is '1'. Obviously, the smaller the block size is, the larger the embedding capacity (*EC*) will be gained, but there was an associated higher error rate in image recovery and data extraction. In [14], Hong *et al.* observed that there was space for improvement in designing the fluctuation function in [13], however, it still does not completely solve the problem of misjudgement. In 2012, Zhang [15] proposed another RDHEI scheme to create a sparse space to accommodate some secret messages by compressing the LSBs of pixels in an encrypted image. Their scheme collected M-LSBs of L pixels and compressed $M \cdot L$ bits into $(M \cdot L - L_S)$ bits using matrix operations, where L_S is the number of bits for secret messages. Thus, Zhang's scheme [15] provided an embedding rate around L_S/L. The larger the L_S and smaller the L, the higher the embedding rate achieved, but with a higher error rate in image recovery. In 2013, Ma *et al.* [16] proposed a novel RDHEI scheme to achieve a high embedding capacity by reserving the room from the LSBs before encryption, and secret messages were embedded into these pre-reserved positions. The maximum embedding rate provided by their scheme is around 0.5 bpp. In 2018, an effective RDHEI scheme based on MSBs prediction was proposed by Puteaux and Puech [17]. In their scheme, the prediction error location map was first built and directly stored into the encrypted image using the operation of prediction error highlighting. The secret messages were embedded into the replaceable MSB of pixels in an encrypted image. The original image can then be recovered at the receiving side. The maximum embedding rate provided by their scheme is close to 1.0 bpp. Unfortunately, there was still an error rate for image recovery and data extraction, with the probability of $1/2^f$, where f is the length of the highlighted flag.

Currently, there are several RDHEI schemes designed to work in the compressed domain, such as JPEG (Joint Photographic Experts Group) [19], VQ (Vector Quantization) [20] and AMBTC (Absolute Moment Block Truncation Coding) [21]. Due to lower storage costs, compressed images are widely used in applications with finite transmission resources. In 2014, Qian *et al.* [22] proposed an RDH scheme in an encrypted JPEG bit-stream. Their scheme first encrypted the original JPEG bit-stream into a properly organized bit-stream structure. The to-be-hidden bits were encoded with error correction codes and embedded into the encrypted bit-stream by slightly modifying the appended bits corresponding to the AC coefficients. On the receiving side, the hidden bits can be correctly extracted by analyzing the blocking artifacts of the neighbouring blocks, and the original JPEG bit-stream can be recovered perfectly. The average embedding capacity provided by Qian *et al.*'s scheme [22] is around 750 bits. In 2016, Qian *et al.* [23] employed a modified encryption method to encipher a bit-stream while keeping the format compliant for a JPEG decoder. The secret messages were concealed into the encrypted bit-stream by compressing the padding bits of the bit-stream. On the receiving side, an iterative recovery mechanism based on blocking artifacts was utilized to restore the original bit-stream. Their scheme provided a larger embedding capacity than previous work [22], with an average *EC* of 1216 bits. In 2017, an RDHEI scheme for encrypted JPEG bit-stream was proposed by Chang *et al.* [24], which is a technique where the usable room was reserved before encryption. They observed that there was redundant space in a biased bit-stream and designed a new lossless compression algorithm, instead of the binary arithmetic coding, to compress the biased bit-stream to pre-reserve the embeddable room. Then, the JPEG bit-stream was encrypted and the secret messages were concealed into the pre-reserved embeddable room. Moreover, the recipient can extract the embedded messages and recover the original bit-stream. The average embedding capacity provide by their scheme is around 1160 bits. In 2018,

Yin *et al.* [25] 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 bitwise XOR operation using the same stream cipher. Thus, the correlation between two quantization levels in an AMBTC compression code was still kept after encryption. Later, both the bitmap replacement scheme when two quantization levels are equal and the prediction error based RDH scheme were applied to vacate the room to embed the secret messages. On the receiving side, the secret messages can be extracted and the original AMBTC-compressed image can be recovered error-free. The average embedding capacity provided by Yin *et al.*'s scheme [25] is about 6081 bits, which is higher than other schemes [22]–[24]. In Yin *et al.*'s scheme [25], some auxiliary information needs to be recorded and is then embedded into the bitmap of the first several triples by sacrificing some embedding space. Also, the original bitmap of the first several triples should be embedded into the remaining triples together with the secret messages. This means that some usable room is wasted. Additionally, we can found that the correlation within the quantization levels is not fully exploited in their scheme and this limits its ability to carry secret messages.

Although these RDHECI methods [22]–[27], [37], [38] well provide content security and privacy-preserving while effectively reducing the bandwidth of transmission. However, all of them have the weakness of the limitation of embedding capacity. To overcome this problem, this paper proposes a novel reversible data hiding scheme in an encrypted AMBTC-compressed image with high embedding capacity. The contributions of this paper are summarized as follow:

1) We propose an AMBTC-based RDHECI scheme, named as O-AMBTC. First, the original image is scrambled in a block-wise manner and compressed by an AMBTC compression technique. Then the derived AMBTC compression codes are encrypted using the operation of value modulation and stream cipher. After the encryption, the correlations between the quantization levels of triples still exist, such that secret messages can be embedded by exploiting the redundant room derived from the quantization levels. Data hiding is then performed with the use of a labelling strategy. It achieves a higher embedding capacity than most existing methods [22]–[27], [37], [38].
2) We propose another RDHECI scheme based on modified AMBTC compression codes, M-AMBTC, where the parities of quantization levels are modified to be in homomorphism. It increases the number of embeddable elements under the same conditions.
3) The proposed schemes offer a high embedding capacity, where average EC is about 4.4 times greater than that of schemes [22]–[27], [37], [38]. It can also recover the original AMBTC-compressed image losslessly.

TABLE 1. Symbols used in this paper and their definitions.

Symbols	Definitions
RDHECI	Reversible data hiding in encrypted compressed image
O-AMBTC	Original AMBTC based RDHECI scheme
M-AMBTC	Modified AMBTC based RDHECI scheme
$I_o=\{(h^k,l^k,bm^k)\}_1^K$	Original AMBTC compression codes
$I_e=\{(he^k,le^k,bme^k)\}_1^K$	Encrypted AMBTC compression codes
$I_m=\{(hm^k,lm^k,bmm^k)\}_1^K$	Marked encrypted AMBTC compression codes

Symbols	Definitions	Symbols	Definitions
K	Number of image blocks	I	Original image
DE	Set of difference error values between he^k and le^k	EC	Embedding capacity
δ	Length of binary code to label pixels in G_1	G_1	Un-embeddable group
λ	Length of binary code to label pixels in G_2	G_2	Embeddable group
N_{ne}	Number of elements in G_1	$M\times N$	Size of an image
N_e	Number of elements in G_2	$m\times n$	Size of image block

The rest of this paper is organized as follows. Section II reviews related works. Section III describes the detailed procedures of the proposed schemes. Experimental results and analyses are given in Section IV. Lastly, our conclusions are presented in Section V.

II. RELATED WORKS

In this section, we introduce the AMBTC compression technique [21] and review a related reversible data hiding scheme in encrypted AMBTC-compressed image [25]. Later, we briefly state a parametric binary tree labeling strategy since it will be used to label the lower quantization levels in our scheme. Moreover, to better present the proposed scheme, the main symbols used in this paper and their definitions are listed in Table 1.

A. AMBTC COMPRESSION TECHNIQUE

AMBTC [21], [39]–[42] is a technique that compresses an image into a version requiring less storage, and has the simplicity in computation compared with other types of image encoding algorithms, such as JPEG. It is used extensively in practical applications, such as mobile terminals and embedded devices which have limited computing resources, and provides a considerable compression ratio (CR) and an acceptable image quality. The detailed process for AMBTC compression is as follows.

Step 1: Divide a gray-scale image I with a size of $M \times N$ into non-overlapping $m \times n$ blocks. Hence, we can obtain $K = \lfloor M/m \rfloor \cdot \lfloor N/n \rfloor$ blocks in total and denote them as $B = \{B^1, B^2, \ldots, B^k, \ldots, B^K\}$. Also, the set of pixels in the k^{th} block B^k is represented as $B^k = \{b_1^k, b_2^k, b_3^k, \ldots, b_{m\cdot n}^k\}$.

Step 2: Calculate the mean value μ^k of all pixel values in the k^{th} block by

$$\mu^k = \frac{1}{m \cdot n} \cdot \sum_{i=1}^{m \cdot n} b_i^k. \tag{1}$$

Step 3: Separate the pixels in the k^{th} block into two sets, S_h, which includes the elements whose values are equal to or higher than μ^k, and S_l, which includes elements whose values are lower than μ^k. Mathematically, do this by:

$$b_i^k \in \begin{cases} S_h, & if\ b_i^k \geq \mu^k, \\ S_l, & if\ b_i^k < \mu^k. \end{cases} \tag{2}$$

Step 4: Derive a triple $(h^k, l^k, bm^k)$by

$$h^k = \frac{1}{h\_num} \cdot \sum\nolimits_{b_i^k \in S_h} b_i^k, \tag{3}$$

$$l^k = \frac{1}{l\_num} \cdot \sum\nolimits_{b_i^k \in S_l} b_i^k, \tag{4}$$

$$bm_i^k = \begin{cases} 1, & if\ b_i^k \in S_h, \\ 0, & if\ b_i^k \in S_l, \end{cases} \tag{5}$$

where h^k is the higher quantization level of the k^{th} block, and l^k is the lower quantization level of the k^{th} block, and bm^k represents the bitmap for the k^{th} block. Here, h_num and l_num are the numbers of elements in the sets S_h and S_l, respectively. They satisfy $h\_num + l\_num = m \cdot n$.

Step 5: Perform Steps 2 to 4 until all blocks have been processed.

Finally, image I is compressed as AMBTC compression codes and it is now represented as $I_o = \{(h^k, l^k, bm^k)\}_1^K$, where $1 \leq k \leq K$. Then, the process of the de-compressing for AMBTC-compressed image is straightforward, that is, reconstruct pixels for each block using

$$\begin{cases} \bar{b}_i^k = h^k, & if\ bm_i^k = 1, \\ \bar{b}_i^k = l^k, & if\ bm_i^k = 0. \end{cases} \tag{6}$$

where $\bar{b}_i^k$ represents the i^{th} reconstructed pixel in k^{th} block.

In addition, we suppose that one pixel consists of 8 bits, therefore, the compression ratio CR can be defined by

$$\mathrm{CR} = \frac{2 \cdot 8 + m \cdot n}{m \cdot n \cdot 8} \times 100\%. \tag{7}$$

B. REVIEW OF YIN et al.'s SCHEME

In 2018, Yin *et al.* [25] proposed a prediction error based reversible data hiding scheme in encrypted AMBTC-compressed images. In their scheme, the AMBTC compression codes were first obtained with the same method as mentioned in Subsection II-A. For each triple, two quantization levels h and l are encrypted by a bitwise-based XOR operation using the same random binary code; and bitmap bm is encrypted in a similar way. Then, the prediction error values are collected by calculating the difference between the encrypted h and l.

After the above procedures, data hiding is performed in two steps. First, 16-bit secret messages are directly replaced as the bitmap in the k^{th} block when its h^k is equal to l^k, and $flag^k$ is set as '1' at the same time; otherwise, $flag^k$ is set to '0'. Additionally, it is easy to find that the distribution of collected prediction error values as they are similar to a Laplace distribution, thus, histogram shifting based reversible data hiding was utilized to conceal secret messages. Here, assume the two peak-bins are P^1 and P^2, and two zero-bins are Z^1 and Z^2, and $Z^1 < P^1 < P^2 < Z^2$. Secret message d^i can be embedded into l^k by changing l^k to $\bar{l}^k$ according to the following equation:

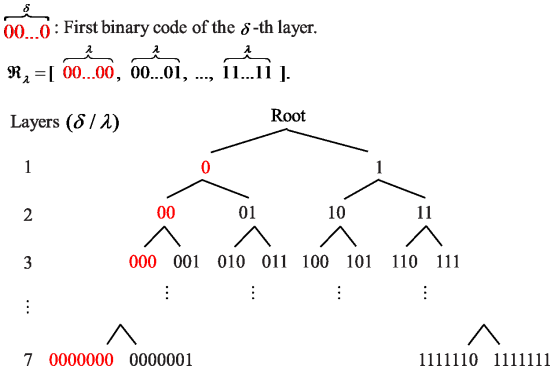


FIGURE 1. Full binary tree T.

$$\bar{l}^k = \begin{cases} l^k + 1, & Z^1 < PE^k < P^1, \\ l^k + d^i, & PE^k = P^1\ and\ flag^k \neq 1, \\ l^k - d^i, & PE^k = P^2\ and\ flag^k \neq 1, \\ l^k - 1, & P^2 < PE^k < Z^2, \end{cases} \tag{8}$$

where PE^k is the prediction error value between two quantization levels in the k^{th} block. In their scheme, some auxiliary information, including two zero-bins, two peak-bins, and a location map that indicates whether the secret message can be embedded into the bitmap in a block, which are recorded and embedded together with the messages.

C. PARAMETRIC BINARY TREE LABELING

In 2019, a parametric binary tree labeling (PBTL) strategy was proposed by Yi and Zhou [18]. Using PBTL, all pixels are labelled into two groups, G_1 and G_2. In their scheme, a full binary tree structure T was designed as shown in Fig. 1, and a tuple parameter (δ, λ) was introduced to provide a series of flexible combinations for labelling strategy, where $1 \leq \delta, \lambda \leq 7$.

For a given δ, the first binary code of the δ^{th} layer, $\overbrace{00\ldots0}^{\delta}$, is chosen to label pixels in group G_1. Notice that the nodes derived from $\overbrace{00\ldots0}^{\delta}$ and all of first nodes in first δ layers should be ignored in the following steps. For G_2, all pixels are further classified into N_λ sub-groups according to the following equation:

$$N_\lambda = \begin{cases} 2^\lambda - 1, & if\ \lambda \leq \delta, \\ (2^\delta - 1) \cdot 2^{\lambda - \delta}, & if\ \lambda > \delta. \end{cases} \tag{9}$$

$\delta=2$	G_1	G_2
$\lambda=1$	00	1

$\mathbb{C}_2^1 = \{1\}$

$\delta=2$	G_1	G_2		
$\lambda=2$	00	01	10	11

$\mathbb{C}_2^2 = \{01, 10, 11\}$.

$\delta=2$	G_1	G_2				
$\lambda=3$	00	010	011	100	101	110	111

$\mathbb{C}_2^3 = \{010, 011, 100, 101, 110, 111\}$.

...

$\delta=2$	G_1	G_2		
$\lambda=7$	00	0100000	...	1111111

$\mathbb{C}_2^7 = \{0100000, 0100001, ..., 1111111\}$.

FIGURE 2. Example of combinations of PBTL when $\delta = 2$ and $\lambda = 1$ to 7.

For each sub-group, the same λ-bit binary code is used to label its elements. For different sub-groups, different λ-bit binary codes are utilized to label the corresponding pixels, respectively. In detail, an ordered set $\mathbb{C}_\lambda^\delta$, includes N_λ different λ-bit binary codes, which is determined by

$$\mathbb{C}_\delta^\lambda = \begin{cases} \Re_\lambda\{2 : 2^\lambda\}, & if\ \lambda \le \delta, \\ \Re_\lambda\{2^{\lambda-\delta}+1 : 2^\lambda\}, & if\ \lambda > \delta, \end{cases} \tag{10}$$

where $\Re_\lambda$ is an ordered set, in which its elements consist of all binary codes in the λ^{th} layer by traversing from left to right side, as shown in Fig. 1. For example, when $\lambda = 3$, thus, $\Re_3 = \{$'000', '001', '010', '011', '100', '101', '110', '111'$\}$. Correspondingly, $\Re_\lambda\{x : y\}$ represents an ordered sub-set of $\Re_\lambda$ and its elements cover the x^{th} code to the y^{th} one within the set $\Re_\lambda$. Fig. 2 gives an example of one combination of the PBTL strategy. Because $\delta = 2$, a 2-bit binary code '00' is used to label pixels in G_1. Also, Fig. 2 lists the N_λ different λ-bit binary codes that are used to label N_λ sub-groups under the different λ. For instance, when λ is 3, the ordered set $\mathbb{C}_2^3 = \{$'010', '011', '100', '101', '110', '111'$\}$ is determined and exploited to label $N_\lambda = 6$ subgroups.

In this paper, all of the lower quantization levels are separated into two groups, G_1 and G_2. G_1 includes un-embeddable pixels and G_2 consists of embeddable pixels. Then, one of the combinations of the PBTL strategies is utilized to label all lower quantization levels. Details of this process are described in Subsection III-B.

III. PROPOSED SCHEMES

This section details the proposed high capacity reversible data hiding scheme for encrypted AMBTC-compressed images. First, the original image is scrambled in a block-wise manner and the scrambled image is then compressed as AMBTC compression codes. Later, the AMBTC compression codes are encrypted by using the methods of value modulation and stream cipher. The encrypted AMBTC compression codes retain the correlations between the two quantization levels in a triple. Then, the secret messages are encrypted and embedded into the redundant room vacated from lower quantization levels. Finally, the extraction of the secret messages and image recovery can be achieved on the receiving side. In addition, another RDHECI scheme based on modified AMBTC compression codes is proposed to further enhance the embedding capacity. Fig. 3 shows a flowchart of the proposed schemes.

A. GENERATION OF ENCRYPTED AMBTC COMPRESSION CODES

In Yin *et al.*'s scheme [25], for each block, its higher quantization level and lower quantization level are encrypted with the same binary number by conducting an XOR operation, attempting to make the prediction error values between them well kept. However, this mechanism is not advantageous for maintaining the natural relationship between the higher quantization level and lower quantization level, resulting in a significant decrement of the frequency of the peak-bin once the prediction error values of the encrypted image are calculated in the histogram.

Our approach is different from the Yin *et al.*'s scheme, as the combination of block scrambling, value modulation, and stream cipher are included during encryption of the AMBTC compression codes. First, block scrambling is used to scramble the image in a block-wise manner. That is, in a one-to-one manner, a 1-D block mapping sequence [28] is generated by encryption key K_p, which is used to obtain a scrambled version of the original image. Then, the scrambled image is compressed as the AMBTC compression codes. Next, for the k^{th} triple (h^k, l^k, bm^k)in I_o, the encryption by value modulation and stream cipher is defined as:

$$\begin{cases} (he^k, le^k) = (h^k + v^k, l^k + v^k) \bmod 256, \\ bme_i^k = bm_i^k \oplus r_i^k, \quad i = 1, 2, \ldots, m \cdot n, \end{cases} \tag{11}$$

where v^k is a random integer generated by an encryption key K_{e1} and ranges from [0, 255], and r_i^k is a pseudo-random binary bit generated by another encryption key K_{e2}. Note that $K_{e1} = hash(I_o$, random number), where $hash(*)$ is a hashing function. Also, for convenience, we denote the encrypted AMBTC compression codes as $I_e = \{(he^k, le^k, bme^k)\}_1^K$, where K is the number of image blocks.

In a few cases, the operation of value modulation may also reduce the natural relationship between the higher quantization level and lower quantization level. However, experimental results in [18] show that the distribution of difference error values produced from the quantization levels is well kept after encryption. This means that the adverse effects of value modulation can be neglected. Our experimental results presented in Subsection IV also verify this view. Therefore, our proposed schemes are able to achieve a high embedding capacity.

B. DATA HIDING PHASE

When a gray-scale image I with a size of $M \times N$ is encrypted as mentioned in Subsection III-A and the secret messages are encrypted by a data hiding key K_h, the process of data hiding can be done per the following procedures.

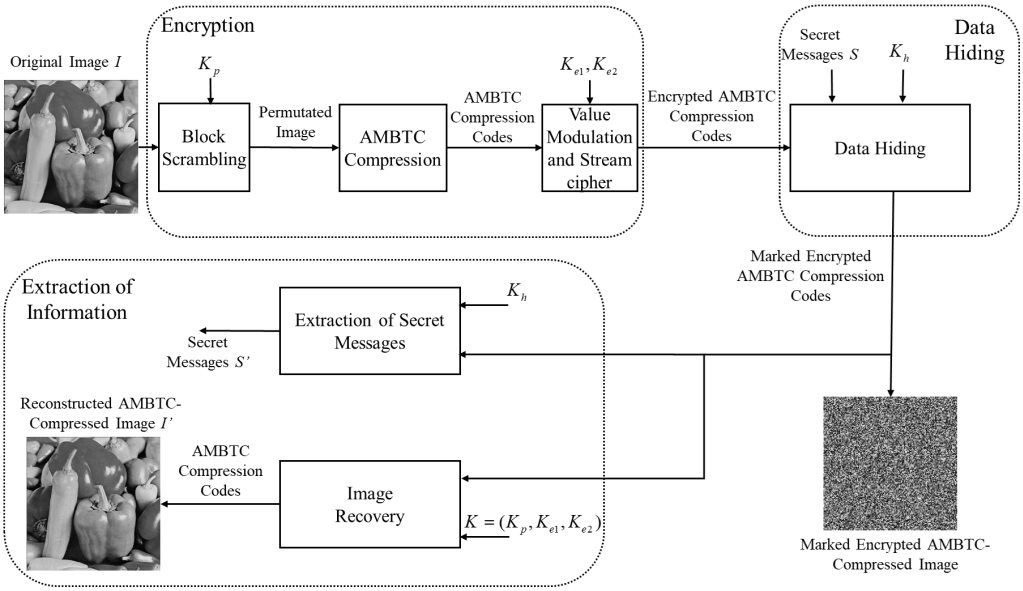


FIGURE 3. Flowchart of proposed scheme.

Inputs: Encrypted AMBTC compression codes $I_e = \{(he^k, le^k, bme^k)\}_1^K$, a tuple parameter (δ, λ), encrypted secret messages Se.

Output: Marked encrypted AMBTC compression codes $I_m = \{(hm^k, lm^k, bmm^k)\}_1^K$.

Step 1: Generate binary codes for labelling. Determine which binary codes are used to label un-embeddable group G_1 and embeddable group G_2. According to the parameters δ and λ, the binary code $\overbrace{00\ldots0}^{\delta}$ is used to label the values in G_1, and $\mathbb{C}_\delta^\lambda$ includes N_λ different λ-bit binary codes that are produced to label the values in G_2. Here, we denote this as $\mathbb{C}_\delta^\lambda = \{c^1, c^2, \ldots, c^{N_\lambda}\}$.

Step 2: Calculate the difference error values $DE = \{de^k\}$. For each triple (he^k, le^k, bme^k), the de^k is defined by

$$de^k = he^k - le^k, \quad de^k \in [-255, 255]. \tag{12}$$

Step 3: Select the embeddable difference error values. Choose the top several N_λ difference error values, which have the highest frequency, from the histogram of the difference error values. Here, we denote the chosen N_λ difference error values as $DE_{max} = \{de^1_{max}, de^2_{max}, \ldots, de^{N_\lambda}_{max}\}$, where de^i_{max} represents the embeddable difference error value, $1 \le i \le N_\lambda$.

Step 4: Value grouping. Traverse difference error value de^k for each triple. If de^k belongs to DE_{max}, the corresponding lower quantization level le^k in the k^{th} triple is classified into group G_2; otherwise, it belongs to group G_1.

Step 5: Value labelling. For each value in G_1, $\overbrace{00\ldots0}^{\delta}$ is adopted to label it by bit replacement, and other (8 - δ) bits are un-modified. Notice that the δ replaced bits of each value in G_1 should be collected as auxiliary information before replacement because of the requirement for reversibility. For each value in G_2, c^i is adopted to label elements whose corresponding difference error value is equal to de^i_{max}, where λ-bit binary code c^i replaces the λ-bit of the binary representations of those elements, where $1 \le i \le N_\lambda$.

Step 6: Payload embedding. After value labelling, the remaining (8 - λ) bits of the binary representation of each element in G_2 are vacated as redundant room, and the payload is embedded into it by bit replacement. Notice that the first triple is skipped and its lower quantization level is used to conceal the parameters δ and λ. Also, the DE_{max} is embedded into the next $\lceil 9 \cdot N_\lambda / 8 \rceil$ triples. Hence, the payload includes the required auxiliary information and encrypted secret messages. The required auxiliary information contains two parts: some original bits of the lower quantization levels in the first $(1 + \lceil 9 \cdot N_\lambda / 8 \rceil)$ triples, and the replaced original δ bits of each value which belong to G_1.

Step 7: Output the marked encrypted AMBTC compression codes.

Finally, the marked encrypted AMBTC compression codes $I_m = \{(hm^k, lm^k, bmm^k)\}_1^K$ are generated, and the marked encrypted AMBTC-compressed image can be reconstructed as mentioned in Subsection II-A.

In addition, we analyze the embedding capacity EC_δ^λ (bits) under the different parameters δ and λ by

$$EC_\delta^\lambda = (8-\lambda) \cdot N_e - \delta \cdot N_{ne} - 8 \cdot (1 + \lceil 9 \cdot N_\lambda / 8 \rceil), \tag{13}$$

where N_{ne}, N_e are the numbers of effective elements in G_1 and G_2, respectively.

In order to further explain the process of data hiding, an example is provided below to illustrate value labelling and payload embedding when $\delta = 2$ and $\lambda = 3$ in Fig. 4. First, according to parameters δ and λ, the binary codes are

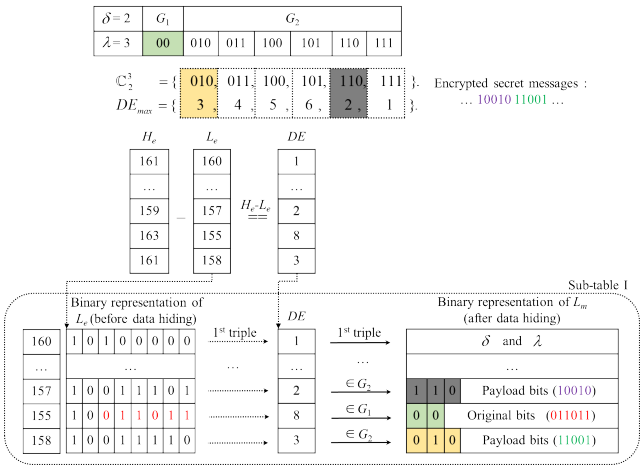


FIGURE 4. Example of value labeling and payload embedding when $\delta = 2$ and $\lambda = 3$.

determined and are shown as a table at the top of Fig. 4. Thus, $\mathbb{C}_{\delta}^{\lambda}$ = {'010', '011', '100', '101', '110', '111'}. Meanwhile, suppose $N_{\lambda} = 6$ embeddable difference error values $DE_{max} = \{3, 4, 5, 6, 2, 1\}$.

Sub-table I in Fig. 4, shows that the first triple is adopted to conceal the parameters δ and λ. Later, as an example, we take the triple (159, 157, *bme*), shown in the 3^{rd} row in sub-table I. Its difference error value $de = he - le = 2$, belongs to DE_{max}. This means that the lower quantization level *le* in this triple is an embeddable element and classified into G_2. Thus, we use the corresponding binary code '110' to replace the first 3 bits of the binary representation of its corresponding $le = 157$, and the other remaining 5 bits are replaced by encrypted secret messages '10010'. Another example is presented in the 4^{th} row in sub-table I, showing when a triple is (163, 155, *bme*), its difference error value is 8, which does not belong to DE_{max}. This means that the corresponding *le* is an un-embeddable element and is classified into G_1. Thus, we should use the binary code '00' to replace the first 2 bits of the binary representation of its corresponding $le = 155$, and the remaining 6 bits are un-changed. Notice that its original 2 bits of the binary representation should be recorded before replacement. Similarly, when the triple is (161, 158, *bme*), its *de* is equal to 3, which belongs to DE_{max}. Thus, the first 3 bits of the binary representation of its corresponding $le = 158$ are replaced with '010' and the remaining 5 bits carry the secret message '11001'.

C. EXTRACTION OF INFORMATION

When the recipient receives the marked encrypted AMBTC-compressed codes I_m, the secret messages can be extracted correctly if the recipient owns the data hiding key K_h, and the original AMBTC-compressed image can be obtained if the recipient holds the encryption keys mentioned in Subsection III-A. If the recipient has all keys, both the secret messages and the original AMBTC-compressed image can be obtained successfully.

1) EXTRACTION OF SECRET MESSAGES

Firstly, we derive the AMBTC compression codes $I_m = \{(hm^k, lm^k, bmm^k)\}_1^K$. Then, a tuple parameter (δ, λ) and DE_{max} can be extracted from the lower quantization level in the first $(1 + \lceil 9 \cdot N_{\lambda}/8 \rceil)$ triples. According to the pair of parameter (δ, λ), we can classify the remaining lower quantization levels *lm* into G_1 and G_2 by checking its first δ and λ bits. Then, the $(8 - \lambda)$ bits of elements in G_2 are collected sequentially and form as the payload. Except for the first $8 \cdot (1 + \lceil 9 \cdot N_{\lambda}/8 \rceil)$ bits and $\delta \cdot N_{ne}$ bits auxiliary information, the remaining bits are the encrypted secret messages. Finally, using the data hiding key K_h, we can decrypt the original secret messages S'.

2) IMAGE RECOVERY

After deriving the payload, the lower quantization levels in the first $(1 + \lceil 9 \cdot N_{\lambda}/8 \rceil)$ triples can be recovered. Then, we also can recover the first δ-bit of elements in G_1 using the payload. At the same time, the two sets, $\mathbb{C}_{\delta}^{\lambda} = \{c^1, c^2, \ldots, c^{N_{\lambda}}\}$ and $DE_{max} = \{de_{max}^1, de_{max}^2, \ldots, de_{max}^{N_{\lambda}}\}$ are reconstructed, respectively. For each value lm^t in G_2, its corresponding difference error value de^t can be found by

$$de^t = de_{max}^q, \quad ifc^q = lm^t(\lambda), \tag{14}$$

where $1 \le q \le N_{\lambda}$, $1 \le t \le N_e$ and $lm^t(\lambda)$ represents the first λ bits of the binary representation of lm^t. Then the corresponding encrypted value le^t can be recovered by

$$le^t = (hm^t - de^t) \bmod 256. \tag{15}$$

Finally, the encrypted version of lower quantization levels is recovered. The operation of data hiding was only performed on the lower quantization levels, and the higher quantization levels and bitmap are un-modified. Thus, the encrypted version of higher quantization levels and bitmap are the same as that in the marked version, respectively.

At this point, according to the encryption keys K_{e1} and K_{e2}, the original AMBTC compression code for each triple can be decrypted by

$$\begin{cases} (h^k, l^k) = (he^k - v^k, le^k - v^k) \bmod 256, \\ bm_i^k = bme_i^k \oplus r_i^k, \quad i = 1, 2, \ldots, m \cdot n. \end{cases} \tag{16}$$

Finally, the original AMBTC compression codes $I_o = \{(h^k, l^k, bm^k)\}_1^K$ are decoded, and the original AMBTC-compressed image I' can be reconstructed losslessly after conducting the inverse process of block scrambling using K_p.

D. MODIFIED AMBTC COMPRESSION CODES BASED SCHEME

Although the above proposed scheme obtains a considerable embedding capacity, there is room for improvement in increasing the capacity. At the same time, inspired by schemes [43], [44], another modified AMBTC compression codes based RDHECI scheme, M-AMBTC, is designed to work by adjusting the parity of the quantization levels. The modified AMBTC compression codes based scheme makes

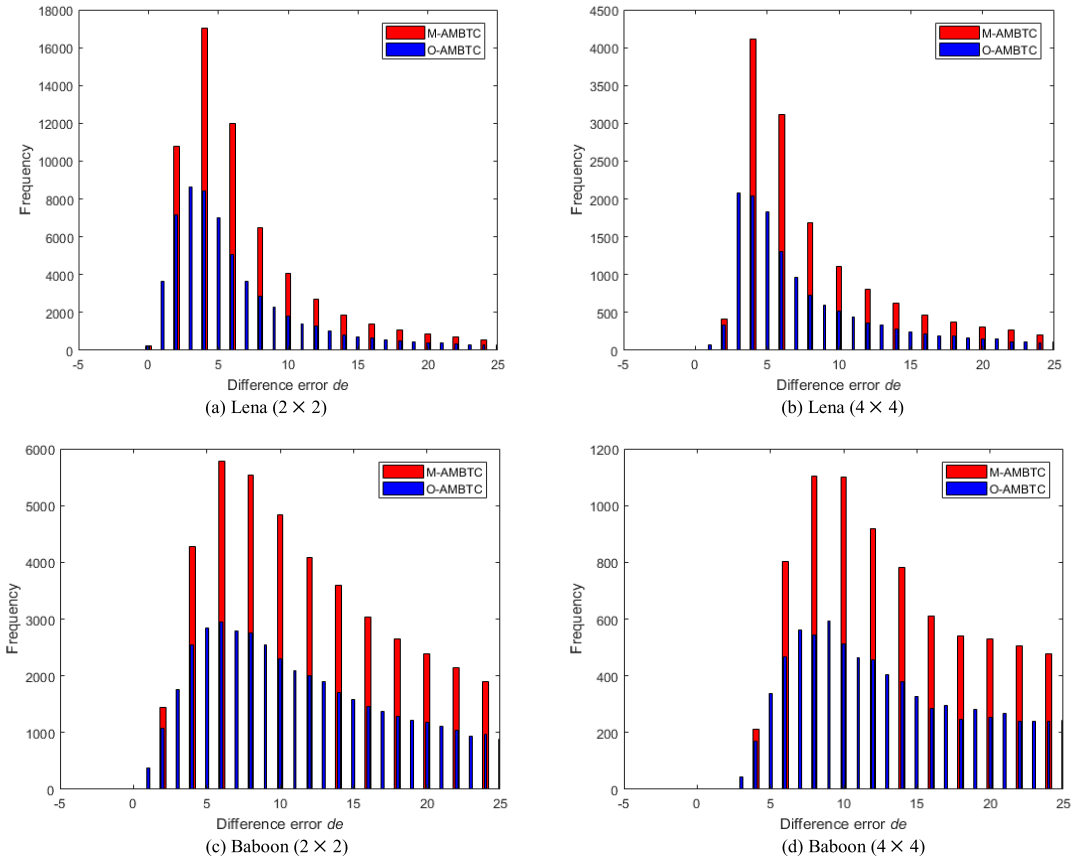


FIGURE 5. Partial histograms of difference error values for M-AMBTC and O-AMBTC: (a), (b) with modifications; (c), (d) without modifications.

TABLE 2. Comparison of AMBTC-compressed image quality for O-AMBTC and M-AMBTC with different block sizes.

PSNRs (dB)				
Block sizes	2 × 2		4 × 4	
Images	O-AMBTC	M-AMBTC	O-AMBTC	M-AMBTC
Elaine	41.8891	41.6015	35.2153	35.1654
Peppers	40.1956	40.0013	33.4276	33.3950
Lena	39.9519	39.7687	33.2281	33.1959
Airplane	39.4733	39.3111	31.9688	31.9459
Boat	39.5449	39.3669	31.8745	31.8506
Man	38.5985	38.4640	32.0794	32.0559
Lake	36.7899	36.6986	29.8819	29.8675
Barbara	32.0456	32.0148	27.0817	27.0740
Baboon	33.7005	33.6549	28.2977	28.2874
Average	**38.0210**	**37.8758**	**31.4506**	**31.4264**

the peaks of the distribution of difference error values significantly high. This means that more elements in all lower quantization levels can be labelled as embeddable under the same conditions, so its embedding capacity is higher than that of the proposed scheme based on O-AMBTC. Details on these modifications are described as follows.

During generation of AMBTC compression codes (h^k, l^k, bm^k)of the k^{th} block, h^k and l^k are modified to be in homomorphism when their parities are different.

Modifications are done thru the following equation:

$$(\hat{h}^k, \hat{l}^k) = \begin{cases} (h^k+1, l^k), & if\,(h^k < 255)\&(h\_num \le l\_num|\ l^k = 0), \\ (h^k, l^k - 1), & if\,(l^k > 0)\&(l\_num < h\_num|\ h^k = 255), \\ (h^k - 1, l^k), & if\,(l^k = 0 \& h^k = 255). \end{cases} \tag{17}$$

In order to analyze the superior performance of the modified AMBTC compression codes, which offers larger embeddable elements than that of O-AMBTC, we calculate the histograms of the difference error values of two quantization levels with or without the modifications operation. The partial statistical results are illustrated in Fig. 5. In the vertical axis, after the homomorphism modifications, the frequencies of even difference error values of M-AMBTC are higher than that of O-AMBTC. Also, in the horizontal axis, since there is no odd difference error value, M-AMBTC can represent and label a wider range of difference error values. Thus, the M-AMBTC scheme has an increased embedding capacity compared to O-AMBTC.

In addition, Table 2 shows a comparison of the results for O-AMBTC and M-AMBTC with respect to the AMBTC-compressed image quality for different block sizes. From the

FIGURE 6. Nine 512 × 512 grayscale images.

results, we can find that M-AMBTC also achieves satisfiable image quality when block sizes are set to 2×2 and 4×4, with average *PSNRs* of 37.8758 dB and 31.4264 dB, respectively. There are a little lower than O-AMBTC, with differences about 0.1453 dB and 0.0242 dB, respectively. In summary, M-AMBTC can also maintain good image quality that is similar to O-AMBTC.

IV. EXPERIMENTAL RESULTS

This section provides the results for several experiments done with the proposed schemes (M-AMBTC and O-AMBTC) and Yin *et al.*'s scheme [25], to demonstrate the effectiveness and superiority of our schemes. All experiments were done with Matlab 2017a on a personal PC with an Intel®Core (TM) i7-3770 CPU @ 3.4 GHz, 8 GB RAM, and Windows 10 operating system. In addition, the nine 512×512 gray-scale images that were used in this paper, Baboon, Barbara, Boat, Elaine, Lake, Lena, Man, Peppers, and Airplane, are shown in Fig. 6.

A. PSNR

Similar to previous work, the peak signal-to-noise ratio (*PSNR*) [29], [30] was used to evaluate the difference between an image I and its recovered image. The higher the *PSNR,* the lower difference they are. *PSNR* is defined by

$$PSNR = 10\cdot\log_{10}\left(\frac{255^2}{\frac{1}{M\cdot N}\cdot\sum_{r=1}^{M}\sum_{c=1}^{N}(I_{r,c}-RI_{r,c})^2}\right), \quad (18)$$

where $I_{r,c}$ and $RI_{r,c}$ are the pixel values of the image I and recovered image, respectively.

B. RESULTS of OUR PROPOSED SCHEMES

In this paper, an effective reversible data hiding scheme in encrypted AMBTC-compressed image was proposed. In our experiments, we utilized the AMBTC compression technique to compress an image into non-overlapping blocks in sizes of 2×2, 2×4, 2×8, 4×2, 4×4, 4×8, 8×2, 8×4, and 8×8. In the following section, two aspects of the experimental results and analyses are presented, including security analysis and experimental results.

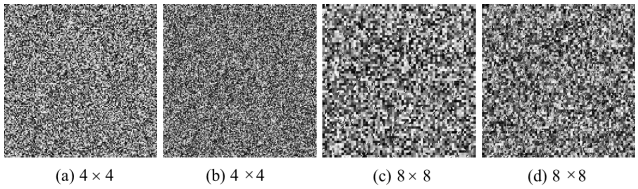


FIGURE 7. The encrypted and marked results of AMBTC-compressed Lena image for various block sizes: (a), (c) encrypted AMBTC-compressed image; (c), (d) marked encrypted AMBTC-compressed image.

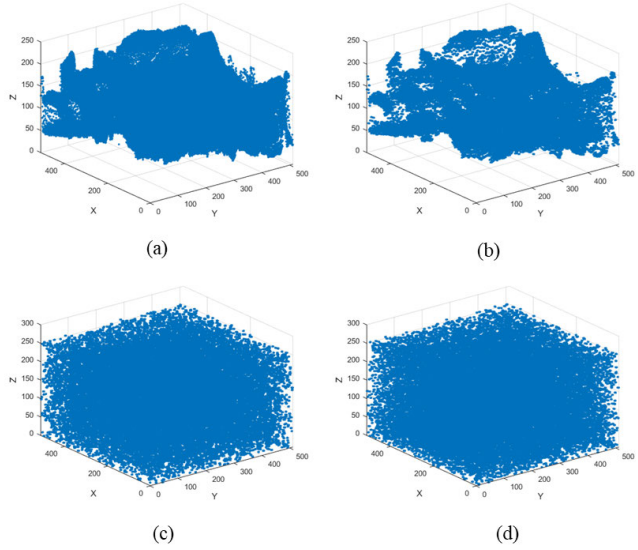


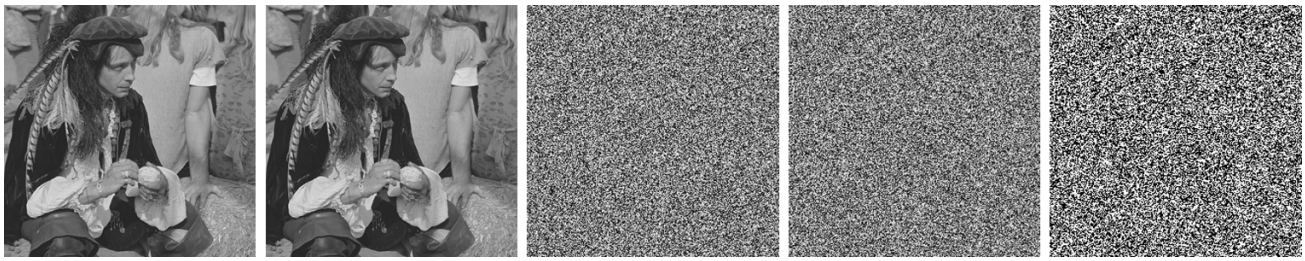
FIGURE 8. Distributions of pixel values for: (a) original Lena image, (b) original AMBTC-compressed Lena image, (c) encrypted AMBTC-compressed Lena image, and (d) marked encrypted AMBTC-compressed Lena image, with a block size of 4 × 4.

1) SECURITY ANALYSIS

a: BRUTE FORCE ATTACK

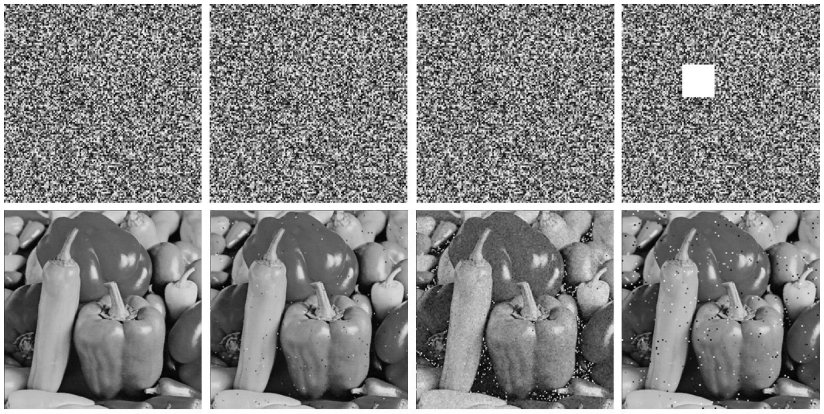
In our approach, the process of generating the encrypted AMBTC compression codes, block scrambling for the original image, and value modulation for the AMBTC compression codes are designed to encrypt the AMBTC-compressed image. At the block scrambling phase, there are $(M\cdot N/m\cdot n)!$ possible permutations in total. Also, for each block, v^k is a random number and ranges within [0, 255], and r^k is a random binary sequence in a length of $m\times n$. Hence, there are also $(256\cdot 2^{m\cdot n})$ possible combinations. In other words, security can be ensured since it is very difficult to decrypt the encrypted AMBTC-compressed image successfully by using only brute force with a probability of $\frac{1}{(256\cdot2^{m\cdot n})^{M\cdot N/m\cdot n}\cdot(M\cdot N/m\cdot n)!}$.

Fig. 7 shows the encrypted and marked results of AMBTC-compressed Lena image for various block sizes. Fig. 7(a) and Fig. 7(c) are the encrypted AMBTC-compressed



(a) AMBTC-compressed image I_{c1} (b) AMBTC-compressed image I_{c2} (c) Encrypted results of I_{c1} (d) Encrypted results of I_{c2} (e) Difference between (c) and (d)

FIGURE 9. Simulation results of chosen-plaintext attack on AMBTC-compressed Man image with a block size of 2 × 2.



(a) Without noise (b) With salt and pepper noise (Density:0.01) (c) With Gaussian noise (Variance:0.005) (d) Cropping with size of 80 × 80

FIGURE 10. Simulation results of noise attacks for the proposed encryption method under the block size of 4 × 4, where the top row shows the encrypted AMBTC Peppers images without or with various noises and the bottom row shows the recovered AMBTC-compressed images.

images Lena without embedding of the secret messages. Fig. 7(b) and 7(d) are the marked encrypted AMBTC-compressed images of Lena that are concealing secret messages. We can observe that the visual contents of the original image were effectively masked after encryption and marking. Fig. 8 shows the distribution of the pixel values of the Lena image at different stages when the block size is set to 4×4. The results show that the pixel values of the original Lena image and original AMBTC-compressed Lena image are all present significantly as one falls and the other rises. However, the distributions of the pixel values of the encrypted AMBTC-compressed Lena image and marked encrypted AMBTC-compressed Lena image are random and uniform, respectively. Although the correlations within two quantization levels after/before the encryption are kept, the spatial relationships between the blocks were broken.

b: STATISTICAL ANALYSIS ATTACK

As we known, image pixels have very strong correlation among each other, hence those images have a relatively lower value of information entropy. Conversely, the more random the distribution of image pixels is, the larger the information entropy is. Table 3 demonstrates the information entropy of the marked encrypted AMBTC-compressed image when the block size is set to 2×2, 4×4, and 8×8, respectively. We can find that the average values of information entropy of these marked encrypted AMBTC-compressed images are 7.9985, 7.9941, and 7.9759, respectively, indicating the correlation among image pixels is destroyed.

TABLE 3. Information entropy results of the marked encrypted AMBTC-compressed images under various block sizes.

Images	Information entropy		
	2 × 2	4 × 4	8 × 8
Elaine	7.9984	7.9945	7.9788
Peppers	7.9984	7.9946	7.9711
Lena	7.9986	7.9934	7.9778
Airplane	7.9985	7.9938	7.9775
Boat	7.9984	7.9943	7.9740
Man	7.9984	7.9944	7.9758
Lake	7.9987	7.9943	7.9778
Barbara	7.9985	7.9939	7.9757
Baboon	7.9985	7.9938	7.9747
Average	**7.9985**	**7.9941**	**7.9759**

c: CHOSEN-PLAINTEXT ATTACK

The chosen-plaintext [18], [30] attack is an attack model for cryptanalysis where the attacker can obtain the ciphertexts for arbitrary plaintexts, and try to gain information that reduces the security of the encryption scheme.

TABLE 4. Embedding capacity *EC* of image Peppers when block size is 2 × 2.

EC (bits)		λ (O-AMBTC)							*EC* (bits)		λ (M-AMBTC)						
		1	2	3	4	5	6	7			1	2	3	4	5	6	7
δ	1	-	43797	105398	163589	163548	118496	60920	δ	1	48912	133537	199430	220279	179504	125105	63434
	2	-	45608	136013	190336	169298	119780	59839		2	-	173544	243190	228688	183348	125896	62767
	3	-	2157	141784	**194034**	167982	118447	58200		3	-	146085	**248992**	227963	182796	125292	61896
	4	-	-	118547	190584	165262	116990	56416		4	-	118626	239156	225536	181705	124406	61011
	5	-	-	95310	181639	161936	114916	54646		5	-	91167	229320	220960	180144	123442	60130
	6	-	-	72073	172694	157602	112608	52831		6	-	63708	219484	216384	178086	122512	59229
	7	-	-	48836	163749	153268	110300	51016		7	-	36249	209648	211808	176028	121442	58328

TABLE 5. Embedding capacity *EC* of image Lena when block size is 2 × 2.

EC (bits)		λ (O-AMBTC)							*EC* (bits)		λ (M-AMBTC)						
		1	2	3	4	5	6	7			1	2	3	4	5	6	7
δ	1	3416	53933	121682	166494	160148	117887	61598	δ	1	70680	137744	212324	215984	179064	126383	64446
	2	-	62576	148291	188086	167083	119920	61159		2	22171	187392	240530	225958	184043	127904	64213
	3	-	21246	151808	**188735**	167046	119717	60048		3	-	161664	**242896**	226661	184764	127997	63776
	4	-	-	129824	184048	163414	118394	58826		4	-	135936	232298	224368	184330	127508	63336
	5	-	-	107840	174286	160104	116960	57484		5	-	110208	221700	219646	183344	127152	62896
	6	-	-	85856	164524	155541	114944	56191		6	-	84480	211102	214924	181686	126592	62456
	7	-	-	63872	154762	150978	112928	54856		7	-	58752	200504	210202	180028	126032	62016

TABLE 6. Embedding capacity *EC* of image Lake when block size is 2 × 2.

EC (bits)		λ (O-AMBTC)							*EC* (bits)		λ (M-AMBTC)						
		1	2	3	4	5	6	7			1	2	3	4	5	6	7
δ	1	-	40	41432	96124	126240	105515	57306	δ	1	5968	59092	123134	173739	162048	118550	62878
	2	-	-	46420	119770	**138188**	108292	57346		2	-	71344	161052	191416	168408	122192	63229
	3	-	-	35224	120688	136998	107172	55708		3	-	31110	165352	**191829**	166962	122842	62588
	4	-	-	-	111784	132523	104756	53671		4	-	-	145061	188016	164793	122204	61866
	5	-	-	-	92989	126024	101021	51604		5	-	-	124770	178750	161776	121433	61132
	6	-	-	-	74194	117201	97448	49450		6	-	-	104479	169484	157422	120440	60398
	7	-	-	-	55399	108378	93245	47152		7	-	-	84188	160218	153068	119111	59664

In this paper, the chosen-plaintext attack has been considered for test robustness of the proposed encryption algorithm. Two original AMBTC-compressed Man images de-compressed from its corresponding AMBTC compression codes are shown in Figs. 9(a) and (b), where their AMBTC compression codes are of one bit difference. Using the same encryption method and random number, the encrypted results of those two images are shown in Figs. 9(c) and (d), and the difference between them is plotted in Fig. 9(e). From Fig. 9(e), we can observe that a slight change in original AMBTC compression codes will lead to a totally different encrypted result, which represents difficult for attackers to gain useful information by analyzing the pairs of plaintexts and its ciphertexts. Therefore, our algorithm is secure against chosen-plaintext attack.

d: NOISE ATTACKS

To further evaluate the robustness of the proposed encryption method, the noise attacks [18], [31] are simulated, including salt and pepper attack, Gaussian attack, and cropping attack. Here, the size of image blocks is set to 4 × 4. The encrypted AMBTC-compressed Peppers images and their corresponding recovered versions are shown in the top row and bottom row in Fig. 10, respectively. From the results, most of the original AMBTC compressed images' information can be recovered.

2) EXPERIMENTAL RESULTS

There are two important tuple parameters, (δ, λ) and (m, n), in the proposed scheme. Different (δ, λ) implies different combinations of labelling strategy, and different (m, n) indicates the use of different block sizes. Tables 4-7 show the performance of the proposed schemes based on two different AMBTC compression codes, i.e., M-AMBTC and O-AMBTC, with respect to the embedding capacity under various (δ, λ) for the marked encrypted AMBTC-compressed images of Peppers, Lena, Lake, and Baboon. The results show that in some cases, the marked encrypted AMBTC-compressed images are unable to embed the secret messages, especially, when λ is set at too small a value or when the image has a complicated texture. Here, '-' represents an un-embeddable case as shown in Tables 4-7.

TABLE 7. Embedding capacity *EC* of image Baboon when block size is 2 × 2.

EC (bits)		λ (O-AMBTC)							*EC* (bits)		λ (M-AMBTC)						
		1	2	3	4	5	6	7			1	2	3	4	5	6	7
δ	1	-	-	2510	38499	69900	79139	49654	δ	1	-	13613	57014	103324	125936	108710	61054
	2	-	-	-	37906	80333	**87968**	47092		2	-	-	65565	120598	143798	113280	62776
	3	-	-	-	21848	76962	85497	44120		3	-	-	52416	119148	**144960**	114682	62104
	4	-	-	-	-	66590	80294	40431		4	-	-	18008	110280	140860	114776	61266
	5	-	-	-	-	53240	73784	36160		5	-	-	-	91297	135168	113782	60412
	6	-	-	-	-	35319	66592	32083		6	-	-	-	72314	127488	112232	59579
	7	-	-	-	-	17398	58532	27760		7	-	-	-	53331	119808	109877	58728

TABLE 8. Embedding capacity *EC* (bits) of images for various (δ, λ).

O-AMBTC										
Images	Block sizes									Condition (δ, λ)
	2 × 2	2 × 4	2 × 8	4 × 2	4 × 4	4 × 8	8 × 2	8 × 4	8 × 8	
Elaine	217785	95736	35884	98319	44578	15856	39153	18705	7389	(3, 4)
Peppers	194034	85908	31747	89205	40028	14610	35877	15821	5716	(3, 4)
Lena	188735	74603	24509	84802	35282	12076	35947	15338	5457	(3, 4)
Airplane	183328	80624	32794	77624	34564	14312	21832	12218	5632	(2, 4)
Boat	166810	65444	23962	61004	25558	9668	18754	7970	2104	(2, 4)
Man	154386	65166	23394	67542	26544	9444	25926	6774	2646	(3, 5)
Lake	138188	57574	16752	57654	24312	6746	21362	8656	2823	(2, 5)
Barbara	92912	35916	11412	38808	13644	4984	14100	5804	1776	(1, 5)
Baboon	87968	39256	15520	33964	14976	3620	12428	3364	-	(2, 6)

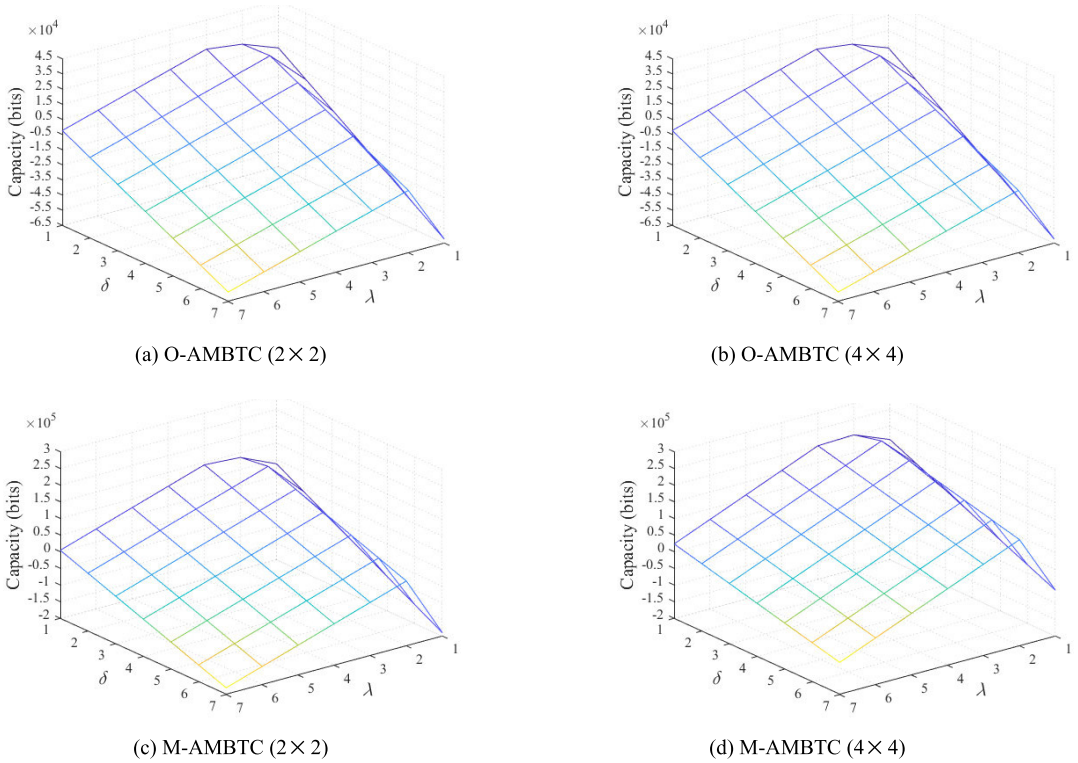
TABLE 9. Embedding capacity *EC* (bits) of images for various (δ, λ).

M-AMBTC										
Images	Block sizes									Condition (δ, λ)
	2 × 2	2 × 4	2 × 8	4 × 2	4 × 4	4 × 8	8 × 2	8 × 4	8 × 8	
Elaine	276048	122936	47792	126048	57752	23072	51592	24768	10056	(3, 3)
Peppers	248992	110040	13112	113688	52584	20160	47816	21472	8272	(3, 3)
Lena	242896	98144	34800	110528	46200	17096	47016	20928	7944	(3, 3)
Airplane	232795	104592	43574	100980	45660	19330	39136	17762	7894	(2, 3)
Boat	217616	87440	31184	82080	33216	13104	25200	10888	3768	(3, 3)
Man	209707	92019	35289	94791	41939	16220	38152	17270	4561	(3, 4)
Lake	191829	81274	29528	80854	34582	12594	30074	12853	4330	(3, 4)
Barbara	132868	49214	14710	55268	21070	6482	19534	7970	550	(2, 4)
Baboon	144960	68586	28950	65220	32244	14232	29868	13800	6060	(3, 5)

In addition, Tables 4-7 show that for different images, the tuple parameter (δ, λ) set to achieve the maximum embedding capacity is different when a block size is 2 × 2. Generally, the smoother the original image, the larger the embedding capacity. Table 4 shows that the comparative smooth image of Peppers obtains a higher *EC*, with maximum *ECs* of 194034 bits and 248992 bits. Correspondingly, in Table 7, the image Baboon with more texture leads to a relatively smaller *EC*, with maximum *ECs* of 87968 bits and 144960 bits. Also, note that M-AMBTC provides a superior payload than O-AMBTC. This is mainly reflected in two aspects: one is that M-AMBTC has fewer un-embeddable cases, and the other is that M-AMBTC gains a larger *EC* than O-AMBTC for the same block size and parameters.

Afterwards, the performance of O-AMBTC and M-AMBTC was analyzed with respect to embedding capacity under various block sizes for the marked encrypted AMBTC-compressed images. Partial experimental results are shown in Tables 8-9 when a determined tuple parameter (δ, λ) is given. Tables 8-9 show that when block size is set to a larger size, the ability of the marked encrypted AMBTC-compressed images to carry a payload decreases. Also, for each image, the embedding capacity reaches the maximum value when the block size is set to 2 × 2.

To further analyze the embedding capacity variation tendency under various combinations of parameter (δ, λ), we performed the experiments on 500 images randomly selected from the UCID datasets [32] to show the average *ECs* provided by O-AMBTC and M-AMBTC when block size is set to 2 × 2 and 4 × 4, respectively. From the results shown in Fig. 11, we can observe that the *ECs* have similar variation tendency for both schemes. Meanwhile, as expected, the proposed M-AMBTC has a higher embedding rate than that of O-AMBTC on average. Concretely, when the block size is 2 × 2, images reach the maximum *EC* of 2.24×10^5 bits for M-AMBTC, which is higher than the maximum *EC* of



(a) O-AMBTC (2 × 2)

(b) O-AMBTC (4 × 4)

(c) M-AMBTC (2 × 2)

(d) M-AMBTC (4 × 4)

FIGURE 11. Average embedding capacity of 500 marked encrypted AMBTC-compressed images for O-AMBTC and M-AMBTC under various combinations of parameter (δ, λ).

TABLE 10. Comparison between M-AMBTC and schemes [18], [35], [36].

Compared list	Scheme [18]	Scheme [35]	Scheme [36]	Proposed scheme (M-AMBTC)
Domain	Spatial domain	Spatial domain	Spatial domain	Compressed domain
Capacity (bpp)	1.52	0.11	0.50	0.80
Number of un-embeddable cases	13.75	-	-	9.25
Encrypted method	Block scrambling, value modulation	Block scrambling, stream cipher	Block scrambling, stream cipher	Block scrambling, value modulation, stream cipher
Brute force attack	Difficult	Difficult	Difficult	More difficult
Chosen-plaintext attack	Yes	No	No	Yes
Nosie attack	Yes	Yes	Yes	Yes
Statistical analysis attack	Theoretically feasible	Theoretically feasible	Theoretically feasible	Yes

Theoretically feasible: It means that the scheme has not been supported by data analysis

1.48×10^5 bits for O-AMBTC. Similarly, when the block size is set to 4×4, the maximum *ECs* provided by O-AMBTC and M-AMBTC are about 25575 bits and 40807 bits, respectively. Obviously, once block size gets large, it makes the decrement in embedding capacity. Unfortunately, it still exists that, in some cases, the marked encrypted AMBTC-compressed images have a poor ability in carrying the secret messages.

We also experimented on 10000/5000 images from the BowsBase datasets [33] and 10000/5000 images from the BossBase datasets [34] to verify the effectiveness of the proposed M-AMBTC scheme. The results are plotted in Fig. 12, where the red line represents the average *EC* of all embeddable images. From Figs. 12(a) and (b), most of images from BowsBase and BossBase datasets are able to embed secret messages when block size is 2×2, and result in average *ECs* of 1.44×10^5 bits and 1.61×10^5 bits, respectively. Figs. 12(c) and (d) show the results performed on every 5000 images from those two datasets when block size is set to 4×4. It is obvious that the proposed M-AMBTC still well done in carrying secret messages.

C. COMPARISONS AND ANALYSES

This paper proposes a reversible data hiding scheme in encrypted compressed AMBTC image, i.e., M-AMBTC. To prove the hiding capacity of our proposed scheme still can compete with most of block encryption-based RDHEI schemes, three representative block encryption-based RDHEI schemes [18], [35], [36] designed for spatial domain are listed in Table 10. From Table 10, we can see even our approach is designed for the compressed domain. The average hiding capacity of our proposed scheme still remains 0.80 bpp, which is only less than that of Yi and Zhou's scheme [18] but significantly higher than that of the rest two schemes [35], [36]. As for the attack analysis, both of our scheme and Yi and Zhou's scheme withstand chosen plaintext and noise attacks. However, our proposed M-AMBTC encrypts the cover image and compression codes by combining the encryption methods of the block scrambling, value

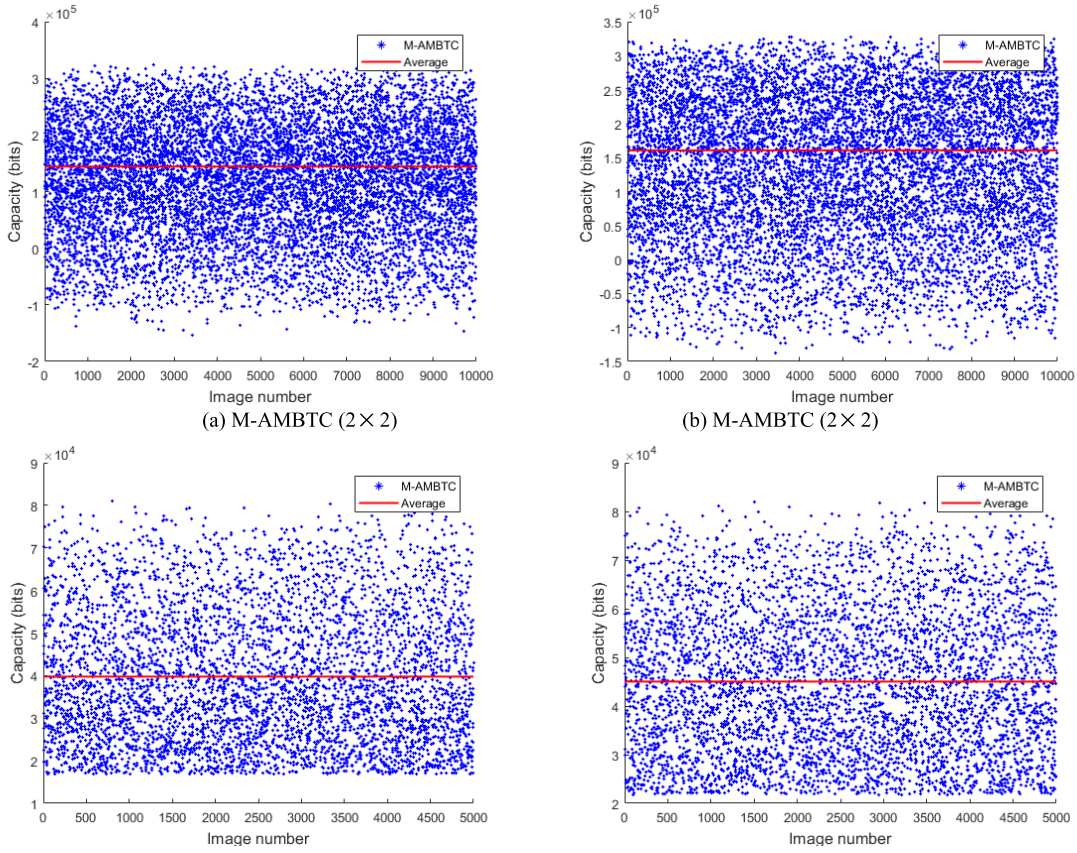


FIGURE 12. Embedding capacity of (a) 10000 images from BowsBase (Average $EC = 1.44 \times 10^5$ bits); (b) 10000 images from BossBase (Average $EC = 1.61 \times 10^5$ bits); (c) 5000 images from BowsBase (Average $EC = 3.98 \times 10^4$ bits); (d) 5000 images from BossBase (Average $EC = 4.50 \times 10^4$ bits).

modulation and stream cipher, making it is more difficult to decrypt the encrypted AMBTC compression codes thru the brute force attack compared with other schemes.

The results provided by the proposed schemes were compared with the results provided by Yin *et al.*'s scheme [25] and are shown in Fig. 13. Fig. 13 shows comparisons of the embedding capacities for the marked encrypted AMBTC-compressed images of Peppers, Lena, Lake, and Baboon. As can be seen, for any block size, the proposed schemes, O-AMBTC and M-AMBTC, achieve significantly superior performance compared to that of Yin *et al.*'scheme [25]. Furthermore, Table 11 provides results for comparisons of the maximum embedding capacities provided by our approaches and the Yin *et al.*'s scheme [25]. When a block size is 2×2, the average *ECs* of O-AMBTC and M-AMBTC are 158166 bits and 210856 bits, which are higher than the average *EC* of 6081 bits provided by [25]. Similarly, when the block size is set to 4×4, averages *ECs* of our approaches are also higher than that of Yin *et al.*'s scheme [25], with differences of 28699 bits and 41211 bits, respectively.

In Yin *et al.*'s scheme [25], it is necessary to record some auxiliary information and embed it into the bitmap of the first several triples by sacrificing some embedding space. Also, the original bitmap of the first several triples will be embedded into the remaining triples together with the secret messages. Moreover, their scheme does not fully excavate and make full use of the characteristic of the AMBTC compression codes. Different from Yin *et al.*'s scheme [25], our schemes further consider the natural correlation of quantization levels within a block and result improved embedding capacity compared to the Yin *et al.*'s scheme [25].

TABLE 11. Comparison of maximum *EC* between our schemes and [25].

Block sizes	2× 2			4× 4		
Images	Yin *et al.*'s scheme [25]	O-AMBTC	M-AMBTC	Yin *et al.*'s scheme [25]	O-AMBTC	M-AMBTC
Elaine	6767	217785	276048	1420	44728	57752
Peppers	5050	194034	248992	926	41080	53234
Lena	6336	188086	242896	1270	36418	48190
Airplane	13370	183328	232795	1365	34564	45660
Boat	5693	166810	217616	932	29382	42490
Man	7658	154386	209707	860	27327	41939
Lake	3445	138188	191829	568	24312	35567
Barbara	4754	92912	132868	524	13644	21967
Baboon	1653	87968	144960	273	14976	32244
Average	**6081**	**158166**	**210856**	**904**	**29603**	**42115**

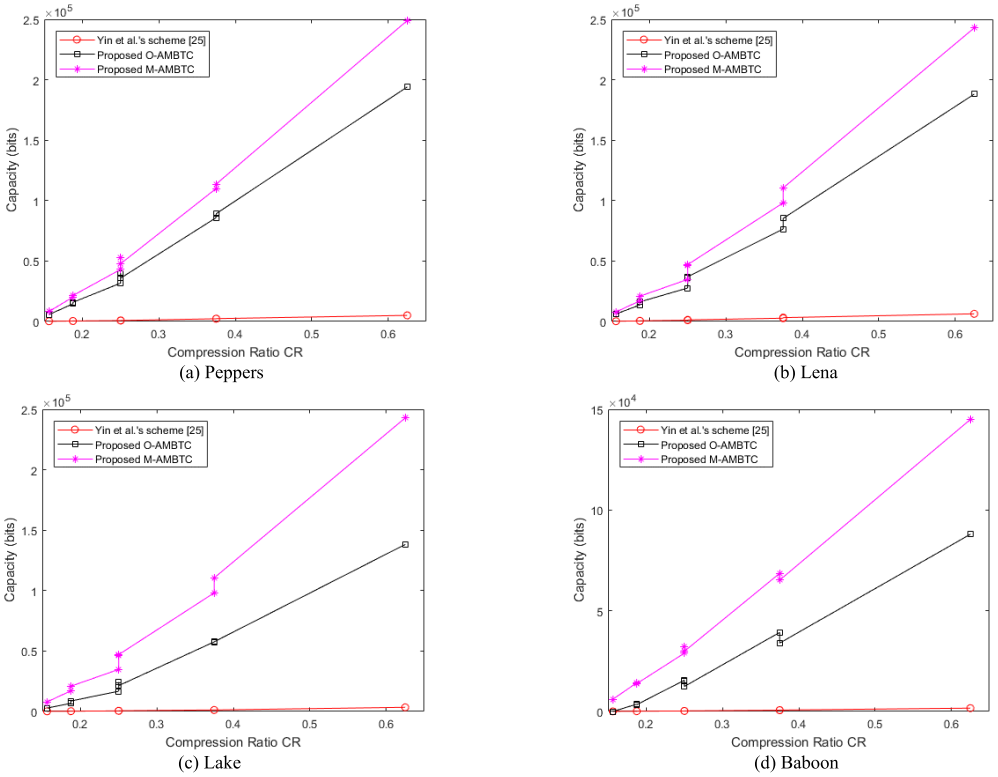


FIGURE 13. Comparisons of *EC* between our schemes and [25] with various CRs.

TABLE 12. Comparison of features for different RDHECI schemes.

Feature	Qian *et al.*'s scheme [22]	Qian *et al.*'s scheme [23]	Chang *et al.*'s scheme [24]	Yin *et al.*'s scheme [25]	Wang *et al.*'s scheme [26]	Nasrullah *et al.*'s scheme [27]	Qin *et al.*'s scheme [37]	He *et al.*'s scheme [38]	Proposed scheme
Compressing method	JPEG	JPEG	JPEG	AMBTC	AMBTC	SPIHT	JPEG	JPEG	AMBTC
Reversibility	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Separable	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Pre-reserving room	No	No	Yes	No	Yes	No	No	Yes	No
Bit error free	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

To further demonstrate the advantages of our proposed scheme, we compared the features and the maximum *EC* between our scheme and other RDHECI schemes [22]–[27], [37], [38] as listed in Tables 12 and 13. It is noted that the schemes of [22]–[24], [37], [38] are reversible data hiding in either an encrypted JPEG bitstream or JPEG images. Similar to [23], both schemes [37], [38] provide a novel JPEG encryption algorithm to encipher a JPEG image and keep the format compliant to JPEG decoders. Qian *et al.*'s scheme [37] designed their data hiding strategy by combining the Huffman code mapping and the ordered histogram shifting, and then achieved an average *EC* of 5108 bits; and He *et al.*'s scheme [38] embedded data into encrypted JPEG images based on invariant zero-run length in the zero-run value pairs, and its embedding capacity is around 6090 bits. Also, scheme of [25] is reversible data hiding in encrypted AMBTC-compressed images, which is under the same constraints as our proposed scheme. In contrast, the scheme of [26] is an adaptive variable N-bit bit-plane truncation image (AVN-BPTI) embedding scheme. AVN-BPTI was exploited to vacate redundant room to conceal secret messages. For each steady block, a prediction error based method was designed to carry more payload. Finally, a chaotic encryption scheme was proposed to enhance the robustness against security vulnerabilities. In Wang et al.'s scheme [26], the operations of encryption and data hiding are not separable, which is quite different from the schemes of [23]–[25], [27], [37], [38] and our proposed schemes. Scheme [27] presented a joint and a separable RDHECI scheme on secure SPIHT (set partition in hierarchical tree) bit stream by reserving rooms through Kd-tree. In their scheme, the secret messages and cover image are extracted without any error. In Table 12,

TABLE 13. Comparison of maximum *EC* for different RDHECI schemes.

EC_{max} (bits)	Qian *et al.*'s scheme [22]	Qian *et al.*'s scheme [23]	Chang *et al.*'s scheme [24]	Yin *et al.*'s scheme [25]	Wang *et al.*'s scheme [26]	Nasrullah *et al.*'s scheme [27]	Qin *et al.*'s scheme [37]	He *et al.*'s scheme [38]	Proposed O-AMBTC	Proposed M-AMBTC
Peppers	750	>=1026	960	5050	22142	47192	4253	-	194034	248992
Lena	750	1364	798	6336	22287	47188	3667	5660	188735	242896
Airplane	-	>=1023	-	13370	23610	48120	-	-	183328	232795
Lake	750	1364	1032	3445	22050	47236	4964	-	138188	191829
Baboon	750	1364	1555	1653	-	48320	7547	6520	87968	144960
Average	**750**	**1228**	**1086**	**5062**	**22522**	**47611**	**5108**	**6090**	**158451**	**212294**

we gived a comparison of features for the proposed scheme and that of other work [22]–[27], [37], [38]. Table 12 shows that the main differences from Wang et al.'s scheme [26] with our proposed scheme are that it does not adopt a pre-reserving room strategy for data embedding, and our approach is a separable scheme.

Table 13 demonstrates that on average, the embedding capacity of our proposed scheme is close to 4.4 times higher than that of [22]–[27], [37], [38]. In other words, our proposed scheme outperforms other schemes in embedding capacity.

V. CONCLUSION

In this paper, we proposed a novel RDH scheme for an encrypted AMBTC-compressed image. The experimental results and analysis demonstrated the superior performance of the proposed scheme. The main contributions of this paper are as follows.

1) An RDHECI scheme based on original AMBTC technique is developed, called O-AMBTC. Concretely, the correlations between quantization levels of a triple were still kept after the encryption methods of block scrambling, value modulation and stream cipher, allowing us to vacate some redundant room to embed secret messages. It provided a higher embedding capacity than most existing methods [22]–[27], [37], [38] (See Table 13).
2) In the meantime, a modified AMBTC compression code based scheme, M-AMBTC, was further applied to enhance the ability to carry secret messages. It is applicable to encrypted AMBTC compressed image with higher embedding capacity, which is 4.4 times greater than that of state-of-the-art works [22]–[27], [37], [38] (See Table 13).
3) Additionally, both O-AMBTC and M-AMBTC are reversible schemes, that is, the secret messages can be extracted error free and the original AMBTC-compressed image can be recovered losslessly (See Table 12).

In the further, the work will be directed towards the development of RDH in encrypted JPEG images or secret sharing in encrypted dual-compressed-images. It may be an interesting problem.

REFERENCES

[1] C.-C. Chang, M.-S. Hwang, and T.-S. Chen, "A new encryption algorithm for image cryptosystems," *J. Syst. Softw.*, vol. 58, no. 2, pp. 83–91, Sep. 2001.

[2] S. Li and X. Zheng, "On the security of an image encryption method," in *Proc. Int. Conf. Image Process.*, Rochester, NY, USA, vol. 2, Sep. 2002, p. 2.

[3] W. Lu, L. He, Y. Yeung, Y. Xue, H. Liu, and B. Feng, "Secure binary image steganography based on fused distortion measurement," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1608–1618, Jun. 2019.

[4] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Process.*, vol. 90, no. 3, pp. 727–752, Mar. 2010.

[5] M. Celik, G. Sharma, A. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[6] W. Zhang, X. Hu, X. Li, and N. Yu, "Recursive histogram modification: Establishing equivalency between reversible data hiding and lossless data compression," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2775–2785, Jul. 2013.

[7] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[8] Y. Hu, H.-K. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 250–260, Feb. 2009.

[9] H.-W. Tseng and C.-P. Hsieh, "Prediction-based reversible data hiding," *Inf. Sci.*, vol. 179, no. 14, pp. 2460–2469, Jun. 2009.

[10] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5010–5021, Dec. 2013.

[11] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 6, pp. 906–910, Jun. 2009.

[12] X. Li, B. Li, B. Yang, and T. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2181–2191, Jun. 2013.

[13] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.

[14] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 199–202, Apr. 2012.

[15] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 826–832, Apr. 2012.

[16] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 3, pp. 553–562, Mar. 2013.

[17] P. Puteaux and W. Puech, "An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1670–1681, Jul. 2018.

[18] S. Yi and Y. Zhou, "Separable and reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 51–64, Jan. 2019.

[19] F. Huang, X. Qu, H. J. Kim, and J. Huang, "Reversible data hiding in JPEG images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1610–1621, Sep. 2016.

[20] C.-C. Chang, W.-L. Tai, and C.-C. Lin, "A reversible data hiding scheme based on side match vector quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 10, pp. 1301–1308, Oct. 2006.
[21] M. Lema and O. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Trans. Commun.*, vol. 32, no. 10, pp. 1148–1157, Oct. 1984.
[22] Z. Qian, X. Zhang, and S. Wang, "Reversible data hiding in encrypted JPEG bitstream," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1486–1491, Aug. 2014.
[23] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 6, pp. 1055–1067, Nov. 2018.
[24] J.-C. Chang, Y.-Z. Lu, and H.-L. Wu, "A separable reversible data hiding scheme for encrypted JPEG bitstreams," *Signal Process.*, vol. 133, pp. 135–143, Apr. 2017.
[25] Z. Yin, X. Niu, X. Zhang, J. Tang, and B. Luo, "Reversible data hiding in encrypted AMBTC images," *Multimedia Tools Appl.*, vol. 77, no. 14, pp. 18067–18083, Jul. 2018.
[26] H.-Y. Wang, H.-J. Lin, X.-Y. Gao, W.-H. Cheng, and Y.-Y. Chen, "Reversible AMBTC-based data hiding with security improvement by chaotic encryption," *IEEE Access*, vol. 7, pp. 38337–38347, 2019.
[27] N. Nasrullah, J. Sang, M. Mateen, M. A. Akbar, H. Xiang, and X. Xia, "Reversible data hiding in compressed and encrypted images by using Kd-tree," *Multimedia Tools Appl.*, vol. 78, no. 13, pp. 17535–17554, Jul. 2019.
[28] P. L. Lin, C.-K. Hsieh, and P.-W. Huang, "A hierarchical digital watermarking method for image tamper detection and recovery," *Pattern Recognit.*, vol. 38, no. 12, pp. 2519–2529, Dec. 2005.
[29] H.-Z. Wu, Y.-Q. Shi, H.-X. Wang, and L.-N. Zhou, "Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 8, pp. 1620–1631, Aug. 2017.
[30] S. Yi and Y. Zhou, "Parametric reversible data hiding in encrypted images using adaptive bit-level data embedding and checkerboard based prediction," *Signal Process.*, vol. 150, pp. 171–182, Sep. 2018.
[31] S. A. Parah, F. Ahad, J. A. Sheikh, N. A. Loan, and G. M. Bhat, "A new reversible and high capacity data hiding technique for E-healthcare applications," *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 3943–3975, Feb. 2017.
[32] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," *Proc. SPIE, Electron. Imag. Storage Retr. Methods Appl. Multimed.*, vol. 5307, pp. 472–480, Dec. 2003.
[33] *BowsBase Dataset*. Accessed: Jul. 1, 2019. [Online]. Available: http://bows2.ec-lille.fr/
[34] *BossBase Dataset*. Accessed: Jul. 1, 2019. [Online]. Available: http://agents.fel.cvut.cz/stegodata/
[35] S. Yi, Y. Zhou, and Z. Hua, "Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion," *Signal Process., Image Commun.*, vol. 64, pp. 78–88, May 2018.
[36] F. Huang, J. Huang, and Y.-Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2777–2789, Dec. 2016.
[37] Z. Qian, H. Xu, X. Luo, and X. Zhang, "New framework of reversible data hiding in encrypted JPEG bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 2, pp. 351–362, Feb. 2019.
[38] J. He, J. Chen, W. Luo, S. Tang, and J. Huang, "A novel high-capacity reversible data hiding scheme for encrypted JPEG bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 12, pp. 3501–3515, Dec. 2019, doi: 10.1109/tcsvt.2018.2882850.
[39] R. Kumar, S. Singh, and K.-H. Jung, "Human visual system based enhanced AMBTC for color image compression using interpolation," in *Proc. 6th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Mar. 2019, pp. 903–907.
[40] A. Malik, G. Sikka, and H. K. Verma, "An AMBTC compression based data hiding scheme using pixel value adjusting strategy," *Multidimensional Syst. Signal Process.*, vol. 29, no. 4, pp. 1801–1818, Oct. 2018.
[41] C. Qin, X. Chen, D. Ye, J. Wang, and X. Sun, "A novel image hashing scheme with perceptual robustness using block truncation coding," *Inf. Sci.*, vols. 361–362, pp. 84–99, Sep. 2016.
[42] C. Qin, P. Ji, C.-C. Chang, J. Dong, and X. Sun, "Non-uniform watermark sharing based on optimal iterative BTC for image tampering recovery," *IEEE Multimedia Mag.*, vol. 25, no. 3, pp. 36–48, Jul. 2018.
[43] A. Malik, G. Sikka, and H. K. Verma, "A high payload data hiding scheme based on modified AMBTC technique," *Multimedia Tools Appl.*, vol. 76, no. 12, pp. 14151–14167, Jun. 2017.
[44] A. Malik, G. Sikka, and H. K. Verma, "A high capacity data hiding scheme using modified AMBTC compression technique," *Int. Arab J. Inf. Technol.*, vol. 16, no. 1, pp. 148–155, Jan. 2019.

GUO-DONG SU (Member, IEEE) received the B.S. degree from Quanzhou Normal University, in 2011, and the M.S. degree in communication and information system from Fujian Normal University, in 2014. He is currently pursuing the Ph.D. degree in information engineering and computer science with Feng Chia University, Taichung, Taiwan.

He is currently a Lecturer with the School of Electronics and Information Engineering, Fuqing Branch of Fujian Normal University, chaired and participated in the research of several provincial projects. His research interests include multimedia security, image processing, and digital forensics.

CHIN-CHEN CHANG (Fellow, IEEE) received the B.Sc. degree in applied mathematics and the M.Sc. degree in computer and decision sciences, and the Ph.D. degree in computer engineering from National Chiao Tung University. He was with National Chung Cheng University, from 1989 to 2005. He has been the Chair Professor of the Department of Information Engineering and Computer Science, Feng Chia University, since February 2005. Prior to joining Feng Chia University, he was an Associate Professor with Chiao Tung University, a Professor with National Chung Hsing University, and the Chair Professor of National Chung Cheng University. He has also been a Visiting Researcher and a Visiting Scientist with Tokyo University and Kyoto University, Japan. During his service in Chung Cheng University, he served as the Chairman of the Institute of Computer Science and Information Engineering, the Dean of the College of Engineering, a Provost and an Acting President of Chung Cheng University, and the Director of Advisory Office, Ministry of Education, Taiwan. His current research interests include database designs, computer cryptography, image compression, and data structures. He is also a Fellow of IEE, U.K. He has won many research awards and honorary positions by and in prestigious organizations nationally and internationally. Since his early years of career development, he consecutively won the Outstanding Talent in Information Sciences of the R.O.C., the AceR Dragon Award of the Ten Most Outstanding Talents, the Outstanding Scholar Award of the R.O.C., the Outstanding Engineering Professor Award of the R.O.C., the Distinguished Research Awards of National Science Council of the R.O.C., and the Top Fifteen Scholars in Systems and Software Engineering of the *Journal of Systems and Software*. He received the award in National Tsing Hua University for his B.Sc. and M.Sc. On numerous occasions, he was invited to serve as a Visiting Professor, the Chair Professor, an Honorary Professor, an Honorary Director, an Honorary Chairman, a Distinguished Alumnus, a Distinguished Researcher, and a Research Fellow by universities, such as Hangzhou Dianzi University, and research institutes.

CHIA-CHEN LIN (MIN-HUI LIN) (Member, IEEE) received the Ph.D. degree in information management from National Chiao Tung University, in 1998. She is currently a Professor with the Department of Computer Science and Information Management, Providence University. Since 2018, she has been with the School counselor of Providence University. Her research interests include image and signal processing, information hiding, mobile agent, and electronic commerce. Since 2018, she has been a Fellow of IET. In additions, from 2009 to 2012, she served as the Vice Chairman of the Tainan Chapter IEEE Signal Processing Society. She also serves as an Associate Editor and an Editor for several representative EI and SCIE journals.