A Reversible Data Hiding Scheme with Security Enhancement Using a Secure Bit Selection Strategy for AMBTC-Compressed Images

DOI: 10.54654/isj.v3i23.1063

Nguyen Duc Tuan*

***Abstract*— Reversible Data Hiding (RDH) is an approach that has garnered significant attention from researchers over the past two decades. In this paper, an Absolute Moment Block Truncation Coding (AMBTC)-based RDH approach is proposed. To increase the embedding capacity (EC), the approach utilizes the high mean and low mean values for data hiding. Additionally, a binary matrix from the AMBTC compressed code is employed to embed the given message bits into encrypted information arrays using the XOR (exclusive OR) operator. This mechanism enhances the security of the hidden message through a secure bit selection method. Consequently, the embedding capacity of the proposed RDH scheme can exceed 98,301 bits when an image block of size 4 × 4 pixels is used. Moreover, the visual quality of the stego-images produced by the proposed RDH scheme surpasses that of state-of-the-art RDH approaches.**

***Tóm tắt*— Giấu tin có thể hồi phục là một giải pháp đang nhận được nhiều sự quan tâm và nghiên cứu của các nhà khoa học trong hai thập kỷ vừa qua. Trong bài báo này, một giải pháp giấu tin có thể hồi phục cho ảnh nén AMBTC được giới thiệu. Để tăng dung lượng tin có thể giấu được, giải pháp đề xuất sử dụng các thành phần lượng tử của dữ liệu nén AMBTC để giấu tin. Ngoài ra, ma trận nhị phân của dữ liệu nén AMBTC được sử dụng để che giấu các bit tin bằng cách sử dụng phép toán XOR. Cơ chế này giúp tăng cường an ninh của tin mật bằng cách sử dụng một phương thức chọn bit có độ an ninh cao. Kết quả là dung lượng giấu của giải pháp đề xuất có thể lên đến 98,301 bit khi sử dụng khối điểm ảnh kích thước 4×4. Chất lượng cảm quan của ảnh mang tin cũng tốt hơn so với các phương pháp giấu tin hồi phục liên quan.**

***Keywords*— Reversible Data Hiding; AMBTC compressed images; bit selection strategy; security enhancement.**

***Từ khóa*— Giấu tin có thể hồi phục; ảnh nén AMBTC; kỹ thuật lựa chọn bit; tăng cường an ninh.**

I. INTRODUCTION

Nowadays, with the rapid development of communication technologies and computer networks, many online systems have been developed to support digital transformation and social economic development development. As a result, a large amount of valuable information is being transferred via the Internet. However, this data faces the risk of being attacked, stolen, or modified without adequate protection methods. Therefore, the demand for securely transferring data through common communication channels on the Internet is increasingly high. Over the decades, many encryption methods have been proposed to protect this information. However, transmitting encrypted data might raise suspicion from attackers, prompting them to develop methods to break the encryption. With the support of powerful hardware systems, certain encryption algorithms may become vulnerable to attacks over time. To address this concern, many data hiding methods [1 - 4] have been developed to conceal information within cover media, especially digital images. These data hiding approaches are generally classified into two categories: Reversible Data Hiding (RDH) and Non-Reversible Data Hiding. RDH schemes allow for the recovery of the original image once data extraction is complete. This is particularly

This manuscript was received on October 19, 2024. It was reviewed on December 7, 2024, revised on December 15, 2024 and accepted on December 19, 2024.

* Corresponding author

important in application areas where the integrity of the cover image is critical, such as in military and medical imaging.

Digital images are the most widely used media in online systems. To transmit images effectively, several image compression techniques have been developed. Similar to block-based image compression methods such as Vector Quantization (VQ) and Joint Photographic Experts Group (JPEG), Absolute Moment Block Truncation Coding (AMBTC) is another technique that effectively reduces the communication cost for digital images. To address the advantages of AMBTC-compressed images, a large number of RDH approaches have been introduced [5 - 11]. These AMBTC-based RDH schemes can be categorized into two types: Type-I and Type-II [6]. AMBTC-based RDH Type-I approaches embed the given secret bits directly into the AMBTC-compressed code, which consists of the quantization levels (H, L) and the binary bitmap. This set of three elements is also referred to as a "trio" [12] or "compressed code" [13]. Consequently, the stego AMBTC-compressed codes can be decoded by the standard AMBTC decoder to generate the AMBTC stego images.

Yin et al. [12] introduced an RDH approach that uses the elements L and the binary bitmap B to embed secret bits. In this approach, the embedding is performed using the histogram shifting technique. First, a stream cipher is applied to the quantization levels H and L, generating a Prediction Error (PE) array. A PE value is calculated as the difference between H and L. The encryption process creates space for embedding the message bits. Following this, a histogram of the Prediction Error (PE) is generated. The data bits are concealed into L based on the correlation between PE and the pairs of Peak points (P_1, P_2) and Zero points (Z_1, Z_2). However, using only the low mean values (L) to carry message bits limits the embedding capacity (EC) of this approach. To enhance EC, Wu et al. [14] developed a Partial Reversible Data Hiding scheme. This approach directly embeds two message bits into each element H or L. Additionally, 6 bits are further concealed in the binary bitmap B using a Hamming-based data hiding method to increase EC. Nevertheless, modifying the binary bitmap of the AMBTC-compressed code using the Hamming-based bit-flipping method is not reserved. Hui et al. [15] introduced a new RDH approach that employs a threshold to classify trios into complex and simple categories. Different embedding methods are applied to hide the given message based on the complexity characteristics of these trios, thereby enhancing EC. In contrast, instead of using trios to embed secret data, AMBTC-based RDH Type-II approaches [10, 11, 16, 17] hide message bits directly in reconstructed pixels. As a result, the obtained EC is higher than that of Type-I approaches because the number of reconstructed pixels is equal to that of the uncompressed/original image.

However, these approaches face two major disadvantages:

1) The stego AMBTC-compressed codes cannot be decoded by the standard AMBTC decoder, making them incompatible with real applications.

2) The size of the stego AMBTC-compressed codes is equal to that of an uncompressed image, requiring more computer and network resources for transmission.

To address the aforementioned drawbacks of the existing AMBTC-based RDH schemes, Type-I and Type-II, a novel AMBTC-based RDH scheme Type-I is introduced, with its contributions summarized as follows:

- A reversible data hiding algorithm is proposed, allowing the restoration of stego AMBTC images to their original versions after the hidden message is extracted by the extraction algorithm.

- A novel AMBTC-based RDH scheme is proposed, using the RDH embedding algorithm to hide a message bit within two mean values (H and L).

- A secure mechanism is proposed to protect hidden data from extraction attacks, featuring a simple and robust bit selection technique that uses a function with a combination π. The bits selected by this technique are then encrypted by

performing the XOR operation with the given message bits, before embedding the result into the AMBTC-compressed codes.

- An optimized transmission strategy is presented, enabling the sending of stego AMBTC-compressed codes along with an additional encrypted message. The combination π is used as a security key and can be predefined by both the sender and receiver, as its size is 21 bits.

The rest of this paper is organized as following.

A standard AMBTC image compression algorithm and the two previous AMBTC-based RDH approaches are briefly described in Section 2. The proposed AMBTC-based RDH scheme, constructed using the data embedding RDH algorithm and the data extraction and image recovery algorithm, is detailed in Section 3. The experimental results are demonstrated in Section 4. The conclusion of advantages, drawbacks and future work, is drawn in Section 5.

II. Related Work

In this section, at first, a conventional AMBTC image compression algorithm is briefly described. Then, the two previous AMBTC-based RDH approaches are presented.

A. AMBTC image compressed algorithm

An image I with dimensions $HI \times WI$ is segmented into M consecutive blocks, each with size of $r \times r$:

$$M = \left\lfloor \frac{HI}{r} \right\rfloor \times \left\lfloor \frac{WI}{r} \right\rfloor \quad (1)$$

In a block, denoted x_j is an element of the block, where $j = 1,2,\ldots,r^2$ represents the pixel's index assigned from top to bottom, left to right of the considered block. The average value of the all pixels in the block is calculated using the formula:

$$AVG = \frac{1}{r^2}\sum_{j=1}^{r^2} x_j, \quad (2)$$

A bit sequence B is generated based on the correlation between the pixels's value and the estimated average value. If $x_j < AVG$, then B_j = 0, otherwise, B_j = 1. Let q denote the number of bits '1' in B, the two quantities, L and H, are approximately calculated by the following formulas:

$$L = \frac{1}{(r^2-q)}\sum_{B_j=0} x_j, \quad (3)$$

$$H = \frac{1}{q}\sum_{B_j=1} x_j, \quad (4)$$

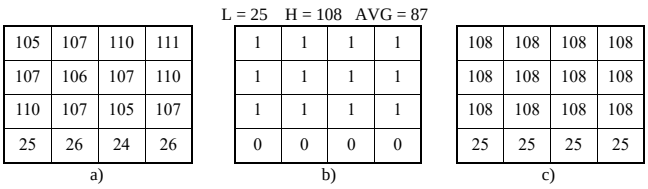

Figure 1. (a) natural image block, (b) ambtc bitmap block, (c) reconstructed image block

Figure 1a illustrates a specific example of an original image block (uncompressed image) with an average value calculated using formula (2). The binary block B determined based on AVG=87.062 is shown in Figure 1b. The values of L and H are calculated according to formulas (3) and (4), respectively. With values L = 25 and H = 108, the corresponding decompressed image block is given in Figure 1c.

B. Reversible data hiding in encrypted AMBTC images

To increase the number of message bits that can be embedded into AMBTC-compressed images, Yin et al. [12] developed an RDH approach using encrypted AMBTC images. At first, the higher mean (H) and lower mean (L) of a triple (i.e., trio) in an AMBTC-compressed image are encrypted by performing the XOR operation between these data and a randomized bit stream. Then, secret data can be embedded into the redundant space created by the image encryption. In this proposed approach, the introduced embedding method employs the prediction error (PE) histogram of the encrypted trio's data to identify which quantization levels (L) are used in data hiding. The prediction error (PE) is defined as follows:

$$PE_i = H_i - L_i, PE_i \in [-255, 255], \quad (5)$$

where H_i and L_i are encrypted elements generated by a stream cipher described in section 3.1 of reference [12]. A histogram of PE is calculated

to identify pairs of peak points (P_1 and P_2) and zero points (Z_1 and Z_2), respectively. These points are required to satisfy this condition: $Z_1 < P_1 < P_2 < Z_2$. The data embedding is involved to solve these following cases.

- **Case 1**. If $H_i = L_i$, then a message bit is hidden directly to L_i. An array *flag* is used to record this case, *flag*(i) = 1 if $H_i = L_i$, otherwise it gains a value of 0.

- **Case 2**. L_i is modified to embed secret bits, therefore the overflow and underflow issues happen. As a result, a location map is employed. If $L_i = 255$ or $L_i = 0$, then the corresponding location map element is set to 1 and L_i is modified to be 254 or 1. If $L_i = 254$ or $L_i = 1$, then the corresponding location map element is set to 0 and L_i remains unchanged.

The given message bits are hidden into elements L_i according to the following equation.

$$L = \begin{cases} L_i + 1, if\ Z_1 < PE_i < P_1 \\ L_i + d, if\ PE_i = P_1\ and\ flag_i \neq 1 \\ L_i - d, if\ PE_i = P_2\ and\ flag_i \neq 1 \\ L_i - 1, if\ P_2 < PE_i < Z_2 \end{cases} \quad (6)$$

where $d \in \{0,1\}$ is the message bits. The array *flag* is utilized for recording whether the bit plane in the i-th triple is replaced by additional data in the embedding process.

The data extraction and image recovery are performed in reversed order. After the hidden data is extracted, the $flag_i$ is employed to restore the AMBTC-image. This approach can employ various sizes of non-overlapping blocks: 2×2, 4×2, 8×2, 2×4, 4×4, 8×4, 8×2, 8×8, to embed the message bits. The EC of this approach can exceed a thousand bits; however embedding auxiliary information, i.e., location maps into trios, leads to the partial recovery of the used AMBTC images. This might create an unintended distortion to original AMBTC-images. Additionally, a sender and receiver should exchange the used data-hiding key and image encryption key. This mechanism is faced with the risk of transferring these keys via the common communication channels and a certain amount of resources is required for performing this task.

C. AMBTC-based RDH approach introduced by Hui and Zhou

To increase the number of message bits that can be embedded into AMBTC compressed codes (trios), Hui et al. developed an RDH approach [15]. In this approach, the *trios* are classified using a threshold *T*: complex *trios* and simple *trios*. The embedding process is performed in three major phases as follows:

1) Preprocessing: in this stage, the temporal value of a quantization level *H*, denoted as H', is calculated by setting its least significant bit (LSB) to '1'. This operation is used to estimate the difference between H' and L: d_i. The original LSBs of elements *H* are stored in an array A_1 to recover these level *H* after message extraction is finished. If $d_i < T$, then the considering *trio* is classified as a simple *trio*. The current element *H*, h_i is modified to mark its simplicity (smoothly) of it. The positions of the simple trios that have been modified will be stored in CS. Similarly, if $d_i \geq T$, the trio will be marked as a complex trio and stored in the *CL* array. Likewise, the LSBs of the *H* components will be stored in A_2 to support the process of data extraction and image recovery.

2) Embedding the given message bits into the simple *trios* using a Huffman code table to reduce the number of required cover bits.

3) Hiding message bits into complex trios: each pair of trios selected from *CL* undergoes a reversible contrast mapping (RCM) transformation to address the overflow issue. In gray-scale images, the overflow problem occurs when a pixel value is 255, and altering it to hide a message bit causes the pixel to exceed 255 (e.g., 256). This issue affects the accuracy of message extraction and original image recovery. After that, the given message bits are concealed into pairs of complex trios using the substitution technique applied to the LSBs of these pairs.

The above phases are performed until there are no message bits left to embed or the embedding process has examined all of the AMBTC trios. The stego trios are then sent to the receiver, along with the used threshold *T* and the two auxiliary arrays, A_1 and A_2. However, the

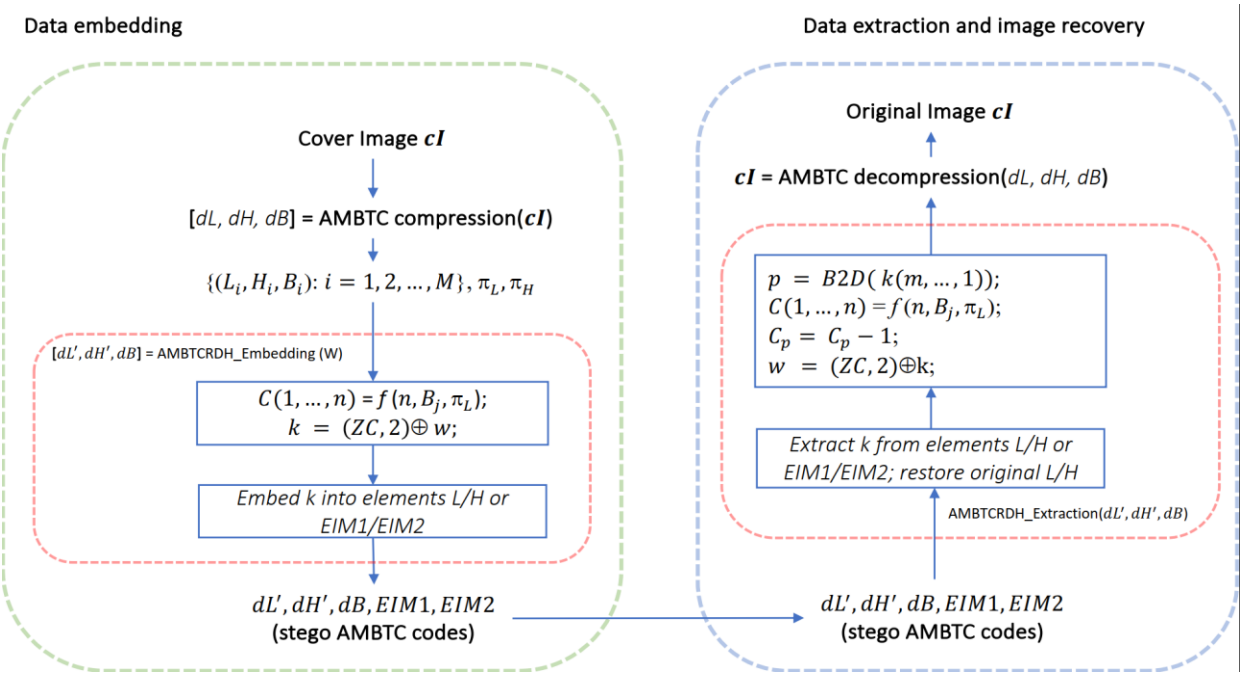

Figure 2. AMBTC-based RDH scheme thay cho AMBTC-based RDH scheme

average embedding capacity (EC) of this approach is 50,000 bits [15]. Additionally, the message embedded could be extracted without any protection if an attacker can obtain the transferred information: the used threshold *T*.

III. Proposed scheme

In this section, the proposed AMBTC-based RDH scheme is presented. To increase the number of message bits that can be embedded into AMBTC compressed code, the proposed scheme utilizes elements of the AMBTC trios (*dL*, *dH*) in data hiding. In a trio, *dL* is an array consisting of elements *L*, *dH* contains the quantization levels *H*, and *dB* is the binary bitmaps. The overall framework of the proposed AMBTC-based RDH scheme is exhibited in Figure 2, where the processes, including data embedding, data extraction and image recovery, are involved. To minimize the distortion to the AMBTC trios, only the quantization levels *L* and *H* are utilized during data hiding.

In the embedding process, the given message is segmented into arrays of size 1 × 3 (*w*). To conceal secret bits without modifying a binary bitmap B_i, bits (randomly selected from B_i) are concealed by performing XOR operation on *w* and (*ZC*,2) to form *k*. The embedding process produces two extra encrypted arrays, *EIM*1 and *EIM*2, when secret data bits are embedded into *H* and *L* elements. *EIM*1 and *EIM*2 can be sent together with stego AMBTC compressed codes securely to a receiver or can be transmitted separately. The detailed description of this mechanism is given in section III.C.

A. Encrypted information and transmitting mechanism

Let $\Omega_{m\times n}$ denote the space of vectors whose components belong to the set {0,1} and have dimensions $m \times n$. When $m = 1$ or $n = 1$, for simplicity, we use the notation Ω_n or Ω_m. Choose *m* and *n* such that $n = 2^m - 1$. We have a matrix $Z \in \Omega_{m\times n}$.

$$Z = \{(i,j) : (i,j) \in \{0,1\}, 1 \le i \le m, 1 \le j \le n\} \quad (7)$$

Denote $Z_1 = (0,..,0,1)^T, Z_2 = (0,..,0,1)^T, ..., Z_m = (0,..,0,1)^T$ is the columns in *Z*.

Consider an array *n* bits $C = (c_1, ..., c_n) \in \Omega_n$.

We have:

$$(ZC, 2) = (\mathrm{Z}(1,1)c_1 + \cdots + \mathrm{Z}(1,n)c_n \bmod 2, .., \mathrm{Z}(m,1)c_1 + \cdots + \mathrm{Z}(m,n)c_n \bmod 2) \quad (8)$$

It is obviously, $(ZC, 2) \in \Omega_m$ Denote $\oplus$ is the XOR operator of the vectors belonging to Ω_m. If both of a and $b \in \Omega_m$ then:

$$a \oplus b = (a_1 \oplus b_1, a_2 \oplus b_2, \dots, a_m \oplus b_m)^T \in \Omega_m \quad (9)$$

For $w \in \Omega_m$ and $e = (ZC, 2) \oplus w \in \Omega_m$.

Suppose that there is the k^{th} column of Z matrix such that $Z_k = e$, $1 \le k \le n$, and C' is obtained from C by performing the k^{th} bit of C: $c'_k = 1 - c_k, C' = (c_1, \dots, 1 - c_k, \dots, c_n)$. We have $(ZC, 2) \equiv w$, when $e = (0, \dots, 0)$ and $e = (ZC, 2) \oplus w$, then $(ZC, 2) \equiv w$.

In the data hiding process, for each binary bitmap $B_i (i = 1,2, \dots, M)$, we have $C(n, r^2)$ is a number of possible selection solutions of unordered selecting n bits from a binary bitmap. As a consequence, the number of solutions for selecting C_1 and C_2 in an ordered manner from a binary bitmap is given by $C(n, r^2) \times C(n, r^2) \times n!$. Therefore, in the first selection, there are $C(n, r^2) \times n!$ ways to choose n distinct positions, with the order of selection differentiating the components, from the r^2 positions in the B block. In the subsequent selection, which is independent of the first, the number of selection solutions is also $C(n, r^2) \times n!$.

For example, when $r = 4$ and $n = 7$, there are approximately 3×10^{15} different ways to create two arrays C from a single binary bitmap B_i. Consequently, the security against message extraction is extremely high. If attackers attempt to identify which bits were selected from B_j, they would need to try up to 3×10^{15} combinations.

Cover bits selecting is handled by a function $f(n, B, \pi)$, which selects n bits from the binary bitmap B of the AMBTC compressed code. These n bits are chosen from B according to the combination π = {n distinct elements chosen from $\{1,2, \dots, r^2\}$}. For example, with $n = 7$, we have $\pi = \{7,1,2,3,4,5,6\}$, and thus, $f(n, B, \pi) = (B_7, B_1, B_2, B_3, B_4, B_5, B_6)$.

As shown in Fig. 3, we have $\pi = \{7,1,2,3,4,5,6\}$ for the 1st selection and $\pi = \{9,10,11,12,13,14,15\}$ for the second one. As a consequence, there are $(C(n, r^2) \times n!)^2$ possible combinations of the two selections.

In general, π is designed to serve as a secure key. Without knowing the value of this key, it is not possible to correctly select the bits from B for data extraction. Even if an attacker identifies that C_1 or C_2 is chosen from a certain binary bitmap, it is not straightforward to estimate C_1 or C_2, which are selected from other binary bitmaps. To determine the value of π, an attacker must try $C(n, r^2) \times n!$ possible outcomes.

As a result, the hidden message cannot be extracted, and the original image cannot be recovered without the sender's intent. Therefore, the data remains secure.

In addition, the sender and receiver can predefined the used π, thus the stego AMBTC compressed codes and extra encrypted information (*EIM*1 and *EIM*2) can be sent together in a secure way. Without knowledge of π, an attacker can not determine which bits of B_i were used during data embedding and decode secret bits.

B. A function for calculating the statistical frequency of occurrence

In the proposed scheme, the message bits are embedded into m pairs $\{a_1, b_1\}, \{a_2, b_2\}, \dots, \{a_m, b_m\}$, which are selected from quantization levels (H and L), satisfying the following conditions:

- frequency $h(a_i)$ of them in trios is highest in local range,
- $h(b_i)$ is as low as possible.

As a consequence, a frequency statistical function is employed. Let's examines the data array: $f = \{f_i : j = 1, \dots, M, f_i \in [0,255]\}$ and a frequency statistical array $hf = \{hf(x)$: frequency of x in $f, x = 0, \dots, 255\}$. Let a and b denote the maximum and minimum point of $hf(.)$, respectively. Assume $hf(b) = 0$. In the cases where $hf(b) > 0$, b and $hf(b)$ are recorded to be used in data extracting. Then, set $hf(b) = 0$.

C. A proposed AMBTC-based RDH embedding algorithm

Denote $\alpha = sign(b - a)$ is the sign function of the result obtained by performing

operation $(b - a)$. If $(b > a)$, then $\alpha = 1$, otherwise -1 is assigned to this parameter.

At first, the maximum point *a* and minimum point *b* of quantization levels *L*, $dL = \{L_1, L_2, \ldots, L_M\}$, is identified using the frequency estimation function (presented in Section III.B). A minimum point *b* corresponds to a value which no quantization level in the given array of quantizers (*H* or *L*). A maximum point *a* corresponds to a value which the maximum number of quantization level in the given array of quantizers.

We have $hL = \{hL(a)$: the frequency of occurrence of a in $dL\}$. In the case when $hL(b) > 0$, i.e there are existing several quantization levels *L* equals to *b*, $hL(b)$ and the corresponding positions of *b* are stored to use in data extracting. Then, set $hL(b) = 0$. The message bits are embedded into L_j as followings: *C* conducted by *n* bits selected from B_j using the function $f(n, B_j, \pi_L)$ to transform the message bits into secure form k_i; *m* bits of k_i is stored to *EIM*1 and L_j is adjusted as $L'_j = L_j + \alpha$ if $L_j \neq \alpha$; otherwise, embed the highest bit of k_j in L_j by $L'_j = L_j + k_j(1)$ and store the lowest $m - 1$ bits of k_j in *EIM*1(*j*).

For better explanation how the embedding process and message extraction, image recovery work, the detailed example is given in two Figures, 3 and 4. Generally, the proposed scheme is designed with the machinist equipped to select the bits from B_j in different ways to increase the security against message extraction attacks. However, in these example, to clarify in illustration, C_1 consists of the bits from 1 to 7 of B_j, whereas C_2 contains bits from 9 to 15 of B_j, respectively.

The data hiding process begins by identifying three corresponding variables, *a*, *b*, and *α*, from the two arrays *dL* and *dH* of AMBTC-compressed codes. Subsequently, the corresponding sets Γ_1 for *dL* and *dH* are created. After generating the two arrays, C_1 and C_2, 3-bit message sets are prepared. A set of *k* is obtained using the formula $k = (ZC,2) \oplus w$, where *w* represents the message bits array.

Data: Cover AMBTC(*dL,dH,dB*) =

$\{(L_i,H_i,B_i) : i = 1,2,\ldots,M\}$, π_L,π_H

Secret bits $W = \{w_1, w_2, \ldots, w_N\}$

Result: Stego AMBTC $(dL', dH', dB) = \{(L'_i, H'_i, B_i): i = 1, 2, \ldots, M\}$, *EIM*1,*EIM*2

1. select *a*, *b* as extrama of $dL = \{L_1, L_2, \ldots, L_M\}$;
2. create set

   $\Gamma_1(a, b) = \{a + \alpha, a + 2\alpha, \ldots, b - 2\alpha, b - \alpha\}$
3. if $hL(b) > 0$ then *record* $(hL(b))$ and set $hL(b) = 0$;
4. $t = 0$;
5. for $j = 1$ to *M*
   a) $C(1, \ldots, n) = f(n, B_j, \pi_L)$;
   b) $w(m, \ldots, 1) = W((j-1) * m + 1, \ldots, j * m)$
   c) $k = (ZC, 2) \oplus w$
   d) if $L_j = a$ then
      d1. $L'_j = L_j + \alpha * k(1)$;
      d2. $EIM1(t + 1,\ldots,t + m - 1) = k(m,\ldots,2)$;
      d3. $t = t + m - 1$;
   e) *else*
      e1. if $L_j \in \Gamma_1(a, b)$ then $L'_j = L_j + \alpha$;
      e2. $EIM1(t + 1, \ldots, t + m) = k(m, \ldots, 1)$;
      e3. $t = t + m$;
6. Apply steps 1-5 to $dH = \{H_1, H_2, \ldots, H_M\}$ to hide message bits to the quantization levels *H*; For each H_j, we can arbitrarily select a set of bits *C* from B_j.

Algorithm 1: A proposed AMBTC-based RDH algorithm for data embedding

According to the example provided for data hiding, the corresponding *k* of C_1 is [0, 1, 1]. The most significant bit of *k* is embedded into *L*. For $L = 197$, and when *L* is equal to α, the resulting value is $L' = 196$. The remaining two bits of *k* are then embedded into *EIM*.

D. A proposed AMBTC-based RDH algorithm for message extraction and image recovery

In this section, the algorithm for message extraction and image recovery is introduced. Generally, both high and low quantization levels are employed to carry message bits with the same operations performed. However, for

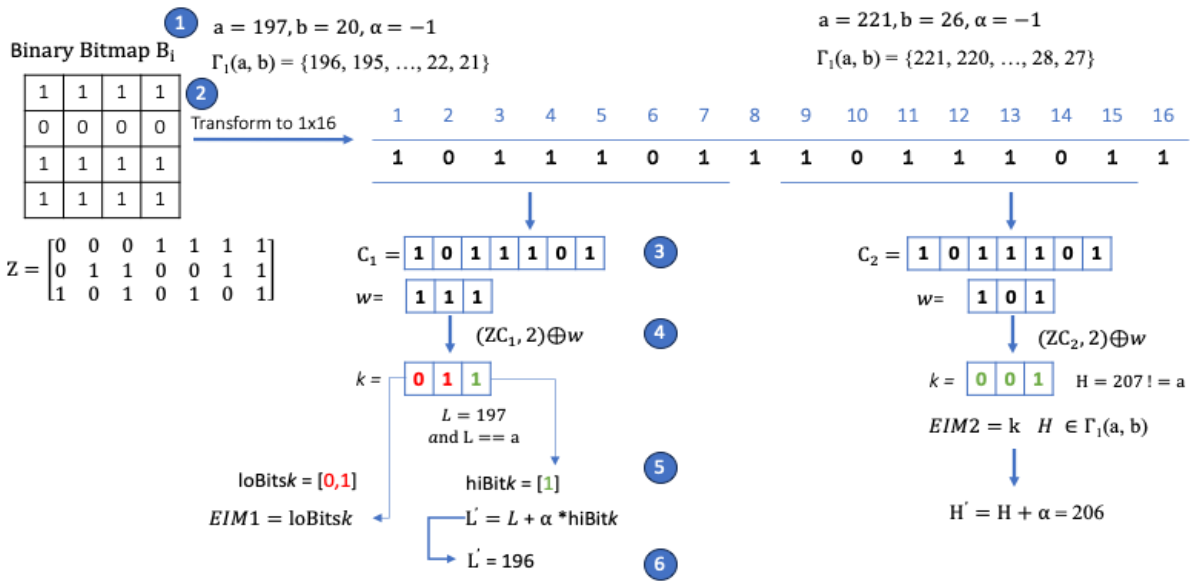

Figure 3. A specific example to illustrate how the proposed scheme embeds message bits into AMBTC compressed codes (H and L)

better clarity in presenting the algorithm, the steps are explained for message embedding using the elements L. In this algorithm, the function *B2D* returns a value that is a decimal number corresponding to its binary form k. This function is utilized to identify a position of bit p, which is altered during the embedding process, to restore its to original values after data extraction is finished.

The main principle of the algorithm is described as follows: if $L'_j \in \{a, a + \alpha\}$, then the highest bit of k is obtained by performing $k(1) = \alpha * (L'_j)$, and the two remaining bits are obtained from *EIM1*; otherwise, all hidden bits are obtained from *EIM1*. In contrast to the embedding process, the p^{th} bit in C is subtracted by 1 before applying the XOR operation with the extracted message bits k as $(ZC, 2) \oplus k$, to obtain the original message bits w.

Data: Stego AMBTC $(dL', dH', dB) = \{(L'_i, H'_i, B_i): i = 1, 2, \ldots, M\}$, $EIM1$,$EIM2$;
Result: Cover AMBTC $(dL, dH, dB) = \{(L_i, H_i, B_i): i = 1, 2, \ldots, M\}$
Secret bits $W = \{w_1, w_2, \ldots, w_N\}$;
1. create a set
$\Gamma_2(a, b) = \{a + 2\alpha, a + 3\alpha, \ldots, b - 2\alpha, b - \alpha, b\}$;
2. $t = 0$;
3. for $j = 1$ to M
a) if $L'_j \in \{a, a + \alpha\}$ then
a1. $k(1) = \alpha * (L'_j - a)$;
a2. $k(m, \ldots, 2) = EIM1(t + 1, \ldots, t + m - 1)$;
a3. $t = t + m - 1$;
a4. $L_j = L'_j - \alpha * k(1)$;
b) *else*
b1. $k(m, \ldots, 1) = EIM1(t + 1, \ldots, t + m)$;
b2. $t = t + m$;
b3. if $L_j \in \Gamma 2(a, b)$ then $L_j = L'_j - \alpha$;
c) $p = B2D(k(m, \ldots, 1))$;
d) $C(1, \ldots, n) = f(n, B_j, \pi_L)$;
e) $C_p = C_p - 1$
f) $w = (ZC, 2) \oplus k$
g) $W((j - 1) * m + 1, \ldots, j * m) = w(m, \ldots, 1)$
4. Restore $L(b)$;
5. Apply steps 1-5 to $dH' = \{H'_1, H'_2, \ldots, H'_M\}$ to extract message bits from the quantization levels H and recover the original pixels;

Algorithm 2: A proposed AMBTC-based RDH algorithm for message extraction and image recover.

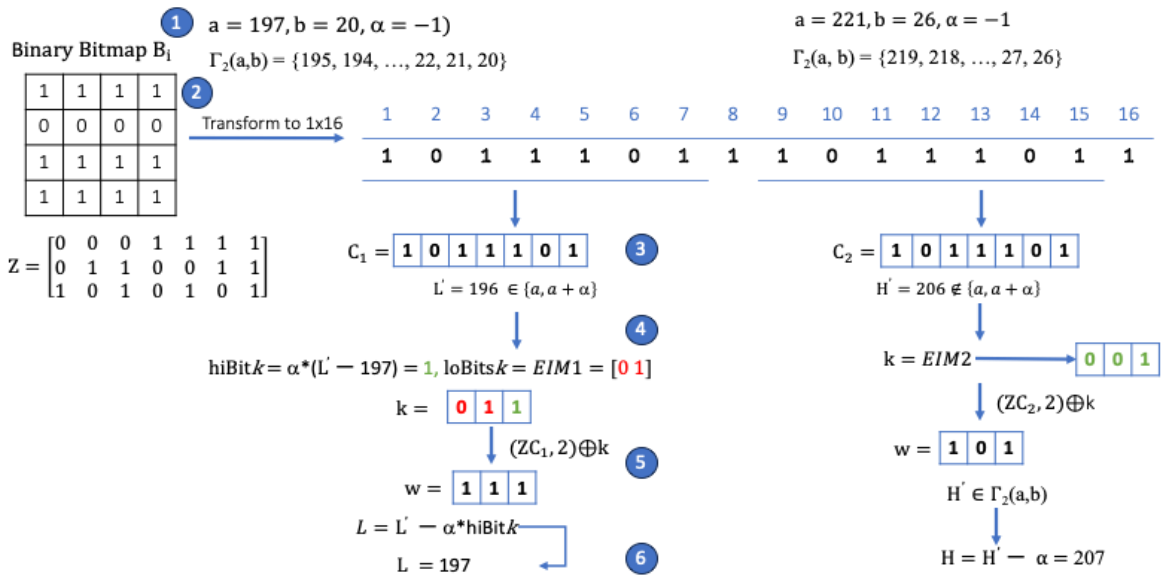

Figure 4. A specific example of message extraction and image recovery performed with stego AMBTC compressed codes (H and L)

Figure 4 illustrates an example of hidden data extraction and image recovery by the proposed scheme. Similar to the example of data embedding, C_1 consists of the bits from 1 to 7 of B_j, whereas C_2 contains bits from 9 to 15 of B_j, respectively.

The data extraction and image recovery process is performed in reversed order compared with the data embedding. At first, the three variables, a, b and α, are calculated from dL', dH', then the corresponding set Γ_2 is created. In this example, with C_1, we have $L'_j \in \{a, a+\alpha\}$, thus the highest bit of k is ($hiBitk = 1$) because $\alpha * (L' - 197)$. The lower bits of k are obtained from the current element of EIM. Then, L is recovered by performing $L = L' - \alpha * hiBitk$. A process to extract message bits, recover the original AMBTC image is executed similarly for the quantization levels H.

IV. Experimental Results and Discussion

To evaluate the performance of the proposed AMBTC-based RDH scheme, we have conducted the experiments with images from SIPI image library [18], BOWS-2 [19] and Kodak [20]. Hardware employed is a notebook with intel i7-10750H processor and 16 GB memory equipped and software is Matlab 2022a running on Windows 11.

Figure 5 illustrates cover images (C) from SIPI image library and their corresponding stego images (S - generated by the proposed scheme) with embedding capacity (EC) is 98,301 message bits.

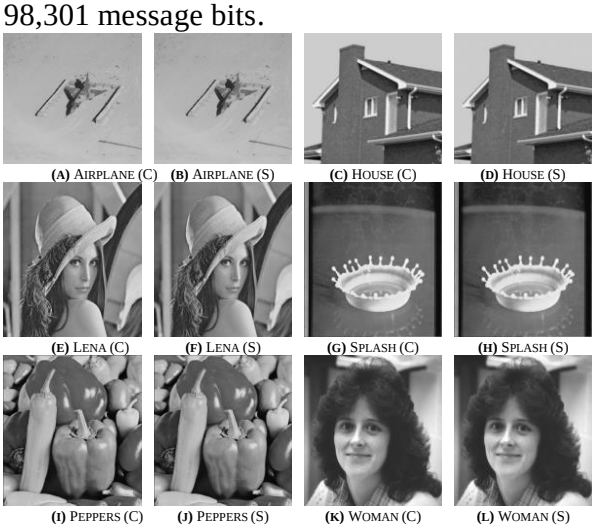

Figure 5. Test images from sipi image library

A. Visual quality discussion

To examine a visual quality of the stego images, Peak Signal to Noise Ratio (PSNR) is employed and it is calculated as follows:

$$PSNR = 10 \times log_{10} \frac{(255)^2}{MSE}, \qquad (10)$$

where max is the highest value of the 8-bits grayscale images [21] and MSE is the mean square error of an image. MSE metric is estimate by performing this following equation.

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (f_{ij} - g_{ij})^2 \qquad (11)$$

where *W, H* are the width and height of an image, *I* and *S* are the cover and stego images, respectively. However, PSNR metric only take an account of the change in value of the pixels.

To further evaluate the performance of the proposed scheme in terms of visual quality, the weighted PSNR (wPSNR) metric is used. This is due to the fact that, unlike the PSNR metric, which relies solely on pixel-by-pixel difference estimation through the Mean Square Error (MSE) [22], wPSNR replicates human vision characteristics by employing the noise visibility function (NVF) [22]. The values of NVF range from [0,1]: extremely textured regions have values close to zero, while flat regions of an image are represented by values near one.

Assume that *I* is an original image and *S* is the corresponding stego-image. The wPSNR metric is measured using Eq. (12):

$$wPSNR = 10\log_{10}\frac{\max(I)^2}{\|NVF(S-I\|^2} \qquad (12)$$

Generally, a high value of wPSNR indicates low visual distortion in the complex regions of the stego-images.

In this experiment, we utilize seven images from SIPI and the experimental results illustrated in Table 1. The first row of Table 1 shows the PSNR values of AMBTC images, which are reconstructed from compressed codes, and original images (uncompressed images). As it can be seen, although the obtained EC of the proposed scheme is highest, the visual quality of it still outperforms the two compared AMBTC-based schemes for several images. This advantage is achieved because the proposed scheme employ the embedding method, which is designed to minimize a distortion to compressed codes (AMBTC *trios*). Meanwhile, in the AMBTC-based RDH approach introduced by Hui et al. [15], several modifications applied to AMBTC compressed codes during the data hiding process. For example, the quantization values *H* are altered to mark the its corresponding trios are simple or complex. This is because, the Hui's approach employs different techniques to embed message bits into simple and complex trios.

Furthermore, as illustrated in Table 1, the wPSNR values of the proposed scheme are closer to those of the corresponding reconstructed images and higher than those of other previous approaches. This indicates that the degradation in texture image regions is minimized. The main reason for this advantage is that the message bits are adaptively embedded into the two quantization levels, *H* and *L*. Meanwhile, the existing RDH approach presented by Hui et al. modifies the quantization *H* to record the classified trios during the data hiding phase. This leads to increased texture distortion (reflected by a lower wPSNR value) in the stego-images.

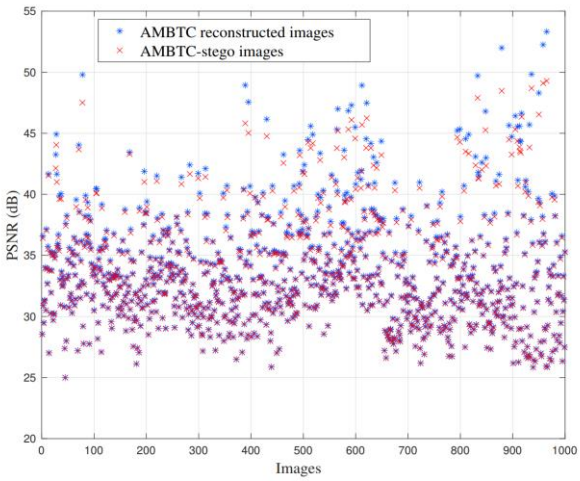

Figure 6. PSNR values of the AMBTC reconstructed images and their corresponding AMBTC stego images

According to the fundamentals of the AMBTC compression technique, the value of high mean *H* is larger than that of the low mean *L* in a trio. Thus, the different value between the two elements, *H* and *L*, is always larger than 0. As a result, in the approach introduced by Yin et al. [12], to yield the space to embed a message bit, the Xor-based image encryption is utilized. The PE values are calculated and then these values are used to form the histogram. The two PP and two ZP are identified to be used in the data hiding process. In this process, the elements *L* are altered to carry message bits or guarantee the accuracy of the data extraction and original AMBTC compressed images recovery. As a consequence, although the EC is quite small, the PSNR is dropped to below 30 dB for most of the used images.

To further examine the perceptual quality of the stego AMBTC images generated by the proposed scheme, 1000 images were randomly selected from BOWS-2. The PSNR values were then calculated for the original uncompressed images and their corresponding AMBTC compressed images. This experiment also estimates the PSNR values of the AMBTC compressed images and their corresponding stego images. These obtained PSNR values are then exhibited in Figure 6. It is obvious that the PSNR values of the stego AMBTC images are nearly to match with those of the cover AMBTC images. This means that the visual degradation of the perceptual quality is minimized. This advantage in terms of visual quality is achieved by utilizing the auxiliary data to hide the secret bits.

B. Embedding capacity discussion

In this experiment, the EC provided by the proposed AMBTC-based RDH approach is discussed. The message bits can be embedded exceed to 98,301 bits. This outperforming is achieved by using the optimized embedding process. In this, elements of the AMBTC compressed code (trio), including the two quantization levels, *H* and *L*, are employed to carry the given message bits. Table 2 demonstrates the obtained highest EC of the proposed scheme and the state-of-the-art methods. The value "N/A" indicates that the corresponding images are not used in the experiments conducted by the compared AMBTC-based RDH methods.

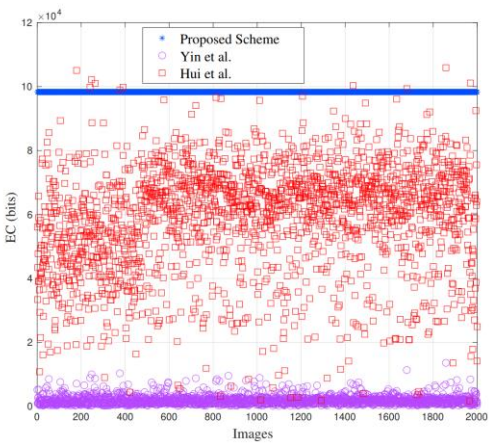

Figure 7. Embedding capacity comparison of the proposed scheme and the existing ambtc-based approaches with 2000 images randomly selected from bows-2

In the method introduced by Wang et al. [24], the low mean and high mean values are used to embed the message bits. These values are first segmented into 128 × 128 macro-block before they are divided into 4 corresponding micro-blocks. Then the given mystery data is hidden into both macro-block and micro-block by a histogram-modified embedding. Therefore, the EC provided by this approach is dependent on the histogram characteristic of these above blocks. Meanwhile, Su et al. [23] present an approach which encrypts the derived AMBTC compression codes. This encryption utilizes 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. Thus, the available EC of this method is limited by the space generated by the used encryption method.

Furthermore, 2000 images were randomly selected from the BOWS-2 dataset to evaluate and compare the embedding capacity (EC) of the proposed scheme and related AMBTC-based reversible data hiding (RDH) approaches. Figure 7 shows the obtained EC (in bits) for the images created by the proposed approach and the two existing methods, Yin et al.'s and Hui et al.'s. The number of message bits that can be embedded using Yin et al.'s method does not exceed 10,000 bits, as only the low mean values *L* of the AMBTC-compressed code are used to carry the secret bits. In contrast, Hui et al. 's approach provides a higher EC compared to Yin et al. 's method. However, the EC achieved by this approach depends on the texture characteristics of the images, specifically the complexity of the AMBTC trios. In the proposed scheme, the EC is steadily achieved at 98,301 bits because each trio and *LM* can carry up to 6 message bits. This result was obtained when pixel block sizes of 4 × 4 were used. Furthermore, the proposed AMBTC-based RDH scheme is designed to work with any size of pixel blocks. Therefore, if pixel blocks size of 2 × 2 are employed, then the EC can be reached to 196,602 bits.

TABLE 1. A COMPARISON OF PSNR (DB) AND WPSNR (DB) WITH THE CORRESPONDING EC (BITS) FOR DIFFERENT SCHEMES AND IMAGES

		Airplane	Cameraman	House	Lena	Milkdrop	Peppers	Woman
	PSNR resonstructed images	37.65	33.92	35.91	33.24	36.82	33.42	38.08
	wPSNR resonstructed images	53.46	49.43	51.30	48.26	52.34	48.88	53.60
Hui et al. [15]	EC	64,904	46,291	49,657	42,664	55,467	41,331	50,108
	PSNR	36.84	32.50	36.00	33.47	36.69	33.88	37.46
	wPSNR	21.73	16.92	23.22	21.53	19.54	21.29	19.75
Yin et al. [12]	EC	3,175	1,413	4,261	1,940	1,940	546	2,650
	PSNR	30.41	23.44	29.31	27.39	27.65	28.90	27.37
	wPSNR	39.82	32.13	38.68	36.90	36.71	39.15	36.20
Proposed scheme	EC	All images achieves an EC is 98,301						
	PSNR	37.47	33.82	35.80	33.18	36.67	33.33	37.67
	wPSNR	43.68	39.20	38.20	38.47	42.31	39.54	42.31

C. Encrypted information analysis

Unlike other AMBTC-based RDH approaches, the proposed scheme does not embed the secret bits directly into the binary bitmap of AMBTC compressed code. Instead, based on the principles of the proposed RDH algorithm, if a mean value $L = a$, the high bit of the message array k is embedded into L. Otherwise, the two remaining low bits of k are concealed in the encrypted information array, *EIM1*. This technique is applied similarly with high mean values H using the corresponding array *EIM*2. The encrypted array k is generated by performing the XOR operator between the message array w and resulting bits extracted from B_j.

TABLE 2. AN EC (BITS) COMPARISON BETWEEN THE EXISTING AMBTC-BASED RDH TYPE-I APPROACHES AND THE PROPOSED SCHEME WITH IMAGES FROM SIPI

	Boat	Airplane	Baboon	Babara	Lena	Man	Peppers	Lake
Su et al. [23] O-AMBTC	29,382	64,904	46,291	49,657	42,664	55,467	41,331	50,108
Su et al. [23] M-AMBTC	42,490	45,660	32,244	21,967	48,190	41,939	53,234	35,567
Wang et al. [24]	22,050	23,610	N/A	N/A	22,287	N/A	22,142	N/A
Proposed scheme	All images achieves an EC is 98,301							

TABLE 3. SECURITY COMPARISON BETWEEN THE PROPOSED SCHEME AND STATE-OF-THE-ART APPROACHES AGAINST DETECTION ATTACKS USING AN ENSEMBLE CLASSIFIER

	OOB Values by Rounds					
Method	**Round 1**	**Round 2**	**Round 3**	**Round 4**	**Round 5**	**Average**
Proposed Scheme	0.160	0,166	0.168	0.167	0.171	0.166
Hui et al. [15]	0.020	0.020	0.010	0.010	0.010	0.014
Yin et al. [12]	0.016	0.018	0.013	0.017	0.015	0.016

D. Security against detection attack by Ensemble Classifier

To further evaluate the security of the proposed scheme against detection attack, a feature-based steganalysis algorithm, the Ensemble Classifier [25], is ultilized in this experiment. The Ensemble Classifier is designed to reduce the computational complexity of existing steganalysis methods, such as those based on support vector machines (SVMs). The out-of-bag (OOB) error, which provides an unbiased estimate of the minimal testing error under equal priors, serves as a key output metric of the Ensemble Classifier. An OOB value close to 0.5 indicates a low likelihood of the Ensemble Classifier detecting the existence of the hidden message.

In this experiment, 2000 randomly selected images are divided into 2 segments with 1000 images for each. One is used to generate stego-images by the proposed scheme and the two previous approaches, Hui et al.'s and Yin et al.'s. These stego-images are then utilized as training data. A remaining set is converted to AMBTC compressed image before being used as testing data for Ensemble Classifier. Then, the features of the training sets and the stegos were extracted by a subtractive pixel adjacency matrix (SPAM) feature extractor [26], in which 686 features collected from an considered image. Each testing set is evaluated five rounds and the output OOB values are presented in Table 3.

It can be observed from Table 3 that the proposed scheme provides significantly higher OOB compared to the other approaches. The main reason for this advantage is that avoiding interference with the image's features helps achieve higher security against statistical steganalysis. This indicates that the proposed scheme outperforms the compared approaches in terms of security against statistical detection attacks. Unlike the watermarking method proposed in [27], which accounts for the robustness of hidden information, this type of secure approach cannot withstand detection attacks as effectively as the proposed scheme.

V. CONCLUSION

In this paper, the proposed AMBTC-based scheme is introduced. It presents the RDH algorithm to hide message bits in AMBTC compressed codes. To enhance the security of the hidden message against extraction attacks, instead of embedding all secret bits into AMBTC trios, these bits are concealed within additional encrypted information using an optimized strategy. This technique uses bits selected from binary bitmaps B_i solely to cover the message bits, with no modifications applied to B_i. A result, the visual quality of the stego AMBTC images is maintained and the EC increases to 98,301 bits when blocks size is 4×4. However, a limitation of the proposed scheme is a magnitude of the extra encrypted data. Therefore, a future work is focusing on minimizing the size of the extra encrypted data.

ACKNOWNLEDGMENT

We appreciate the assistance of our colleagues and the financial support provided by Vietnam National University, Hanoi, School of Interdisciplinary Sciences and Arts.

REFERENCES

[1] Y. Pan, J. Ni, Q. Liu, W. Su, and J. Huang, "Efficient jpeg image steganography using pairwise conditional random field model," *Signal Processing*, vol. 221, p. 109493, Aug. 2024. doi: http://dx.doi.org/10.1016/j.sigpro.2024.109493.

[2] D. Laishram and T. Tuithung, "A novel minimal distortion-based edge adaptive image steganography scheme using local complexity: (beass)," *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 831–854, Sep. 2020. doi: http://dx.doi.org/10.1007/s11042-020-09519-9.

[3] W. Wu and H. Li, "A novel scheme for random sequential high-capacity data hiding based on pvd and lsb," *Signal, Image and Video Processing*, vol. 18, no. 3, pp. 2277–2287, Dec. 2023. doi: http://dx.doi.org/10.1007/s11760-023-02900-9.

[4] R. RoselinKiruba and T. S. Sharmila, "A novel data hiding by image interpolation using edge quad-tree block complexity," *The Visual Computer*, vol. 39, no. 1, pp. 59–72, Oct. 2021. doi: http://dx.doi.org/10.1007/s00371-021-02312-1.

[5] C.-C. Lin, X.-L. Liu, W.-L. Tai, and S.M. Yuan, "A novel reversible data hiding scheme based on ambtc compression technique," *Multimedia Tools and Applications*, vol. 74, no. 11, pp. 3823–3842, Dec. 2013. doi: http://dx.doi.org/10.1007/s11042-013-1801-5.

[6] W. Hong, Y.-B. Ma, H.-C. Wu, and T.-S. Chen, "An efficient reversible data hiding method for ambtc compressed images," *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 5441–5460, Oct. 2016. doi: http://dx.doi.org/10.1007/s11042016-4032-8.

[7] W. Hong, X. Zhou, and S. Weng, "Joint adaptive coding and reversible data hiding for ambtc compressed images," *Symmetry*, vol. 10, no. 7, p. 254, Jul. 2018. doi: http://dx.doi.org/10.3390/sym10070254.

[8] J. Lin, C.-C. Lin, and C.-C. Chang, "Reversible steganographic scheme for ambtc-compressed image based on (7,4) hamming code," *Symmetry*, vol. 11, no. 10, p. 1236, Oct. 2019. doi: http://dx.doi.org/10.3390/sym11101236.

[9] G.-D. Su, C.-C. Chang, and C.-C. Lin, "A high capacity reversible data hiding in encrypted ambtc-compressed images," *IEEE Access*, vol. 8, pp. 26984–27000, 2020. doi: http://dx.doi.org/10.1109/ACCESS.2020.2966234.

[10] J. Lin, S. Weng, T. Zhang, B. Ou, and C.-C. Chang, "Two-layer reversible data hiding based on ambtc image with (7, 4) hamming code," *IEEE Access*, vol. 8, pp. 21534–21548, 2020. doi: http://dx.doi.org/10.1109/ACCESS.2019.2962230.

[11] X. Zhang, Z. Pan, Q. Zhou, G. Fan, and J. Dong, "A reversible data hiding method based on bitmap prediction for ambtc compressed hyperspectral images," *Journal of Information Security and Applications*, vol. 81, p. 103697, Mar. 2024. doi: http://dx.doi.org/10.1016/j.jisa.2023.103697.

[12] Z. Yin, X. Niu, X. Zhang, J. Tang, and B. Luo, "Reversible data hiding in encrypted ambtc images," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 18067–18083, Aug. 2017. doi: http://dx.doi.org/10.1007/s11042-017-4957-6.

[13] R. Kumar and A. Malik, "Multimedia information hiding method for AMBTC compressed images using LSB substitution technique," *Multimed Tools Appl*, vol. 82, no. 6, pp. 8623–8642, 2023.

[14] X. Wu and C.-N. Yang, "Partial reversible ambtc-based secret image sharing with steganography," *Digital Signal Processing*, vol. 93, pp. 22–33, Oct. 2019. doi: http://dx.doi.org/10.1016/j.dsp.2019.06.016.

[15] Z. Hui and Q. Zhou, "A reversible data hiding method for AMBTC compressed image without expansion inside stego format," *Transactions on Internet and Information Systems*, vol. 14, no. 11, 2020.

[16] X. Zhou, G. Yang, W. Hong, K. S. Chen, and T.-S. Chen, "An adaptive data hiding for AMBTC compressed images with recoverability using bound shifting technique," *Multimed Tools Appl*, vol. 82, no. 10, pp. 15593–15612, 2023.

[17] J.-C. Liu, Y. Lin, C.-C. Chang, and C. Chang, "A side match oriented data hiding based on absolute moment block truncation encoding mechanism with reversibility," *Multimedia Tools and Applications*, Jul. 2024. Available: http://dx.doi.org/10.1007/s11042-024-19752-1

[18] A. Weber. (1997) The usc-sipi image database. Signal and Image Processing Institute of the. University of Southern California. Access time: 15/12/2024. https://sipi.usc.edu/database/.

[19] T. P. Patrick Bas, Tomas Filler. (2008, April) Bows-2 image database. Access time: 10/12/2024. http://bows2.eclille.fr/BOWS2OrigEp3.tgz.

[20] Kodak, "The kodak color image dataset," Online, 11, 1999. Access time: 15/12/2024. https://r0k.us/graphics/kodak/.

[21] D. R. I. M. Setiadi, "Psnr vs ssim: imperceptibility quality assessment for image steganography," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 8423–8444, Nov. 2020. doi: http://dx.doi.org/10.1007/s11042-020-10035-z.

[22] A. Saeed, Fawad, M. J. Khan, H. Shahid, S. I. Naqvi, M. A. Riaz, M. S. Khan, and Y. Amin, "An accurate texture complexity estimation for quality-enhanced and secure image steganography," *IEEE Access*, vol. 8, pp. 21613–21630, 2020. doi: http://dx.doi.org/10.1109/ACCESS.2020.2968217.

[23] G.-D. Su, C.-C. Chang, and C.-C. Lin, "A high capacity reversible data hiding in encrypted AMBTC-compressed images," *IEEE Access*, vol. 8, pp. 26984–27000, 2020.

[24] H.-Y. Wang, H.-J. Lin, X.-Y. Gao, W.-H. Cheng, and Y.-Y. Chen, "Reversible ambtcbased data hiding with security improvement by chaotic encryption," *IEEE Access*, vol. 7, pp. 38337–38347, 2019. doi: http://dx.doi.org/10.1109/ACCESS.2019.2906500.

[25] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, Apr. 2012. doi: http://dx.doi.org/10.1109/TIFS.2011.2175919

[26] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010. doi: http://dx.doi.org/10.1109/TIFS.2010.2045842.

[27] A. Kozachok and S. Kopylov, "Robust text watermarking based on line shifting," *Journal of Science and Technology on Information security*, vol. 7, no. 1, pp. 3–13, Apr. 2020. doi: https://doi.org/10.54654/isj.v7i1.50.

ABOUT THE AUTHORS

Nguyen Duc Tuan
Workplace: Vietnam National University, Hanoi, School of Interdisciplinary Sciences and Arts.
Email: tuannguyenduc@vnu.edu.vn
Education: received the PhD degree in Computer Science from Khon Kaen University (KKU), Khon Kaen, Thailand, in 2016. He received the M.SC degree from Le Qui Don Technical Univeristy, Vietnam in 2008.
Recent research interests: Cryptography, Steganography, Reversible Data Hiding, Blockchain.

Tên tác giả: **Nguyễn Đức Tuấn**
Cơ quan công tác: Trường Khoa học Liên ngành và Nghệ thuật, Đại học Quốc Gia Hà Nội
Email: tuannguyenduc@vnu.edu.vn
Quá trình đào tạo: Nhận bằng Tiến sĩ Khoa học máy tính tại Đại học Khon Kaen, Khon Kaen, Thái Lan năm 2016; Nhận bằng Thạc sĩ Khoa học máy tính tại Trường Đại học Kỹ thuật Lê Quý Đôn năm 2008
Hướng nghiên cứu hiện nay: Mật mã, Viết phủ, Giấu tin có thể hồi phục, Blockchain.