symmetry

Article

Detectability Improved Tamper Detection Scheme for Absolute Moment Block Truncation Coding Compressed Images

Wien Hong [1], Xiaoyu Zhou [1], Der-Chyuan Lou [2,3,*], Xiaoqin Huang [4] and Cancan Peng [1]

[1] School of Electrical and Computer Engineering, Nanfang College of Sun Yat-Sen University, Guangzhou 510970, China; wienhong@gmail.com (W.H.); xiaoyuzhou68@outlook.com (X.Z.); anzhitinglan3@outlook.com (C.P.)

[2] Department of Computer Science and Information Engineering, Chang Gung University, Taoyuan 33302, Taiwan

[3] Stroke Center and Department of Neurology, Chang Gung Memorial Hospital, Linkou, New Taipei 20401, Taiwan

[4] Department of Applied Foreign Language, Chengdu Neusoft University, Chengdu 611844, China; huang.xiaoqin@hotmail.com

* Correspondence: dclou@mail.cgu.edu.tw; Tel.: +886-3-211-8800 (ext. 3696)

Received: 4 July 2018; Accepted: 22 July 2018; Published: 2 August 2018

Abstract: Since digital media is gaining popularity nowadays, people are more concerned about its integrity protection and authentication since tampered media may result in unexpected problems. Considering a better media protection technique, this paper proposes an efficient tamper detection scheme for absolute moment block truncation coding (AMBTC) compressed images. In AMBTC, each image block is represented by two quantization levels (QLs) and a bitmap. Requiring insignificant computation cost, it attracts not only a wide range of application developers, but also a variety of studies to investigate the authentication of its codes. While the existing methods protect the AMBTC codes to a large extent, the leakage of some unprotected codes may be insensitive to intentional tampering. The proposed method fully protects the AMBTC codes by embedding authentication codes (ACs) into QLs. Meanwhile, the most significant bits of QLs are symmetrically perturbed to generate the candidates of ACs. The ACs that cause the minimum distortion are embedded into the least significant bits of QLs to minimize the distortion. When compared with prior works, the experimental results reveal that the proposed method offers a significant sensitivity-of-tamper property while providing a comparable image quality.

Keywords: AMBTC; image authentication; tamper detection

1. Introduction

With the rapid growth of the internet, it is getting easier to transmit information through digital media such as images and videos. However, people are getting cautious at the same time since software, such as Photoshop, is making media modification and duplication more feasible as well. Because tampered pictures and videos could result in unexpected problems in aspects of society, business, and international relations, people are paying more attention to the protection of media integrity. Considering that there are massive digital media, it is important to speed up the file-transfer rate and decrease image storage space. Therefore, to achieve less space, digital images are usually stored in either lossless or lossy compressed formats. The lossless compression techniques, such as run-length encoding and Huffman coding, allow a full recovery of original images without causing any distortion. In contrast, lossy compression implies a loss of some information of the original images. However, since the lossy compression can gain a higher compression ratio than the former, it is nowadays commonly used in image compression techniques.

Vector quantization (VQ), joint photographic experts group (JPEG), and block truncation coding (BTC) are typical lossy compression techniques investigated in the literature. Among these methods, the BTC proposed by Delp and Mitchell [1] has an advantage over the others for its relatively low computational complexity and better-reconstructed image quality. Improved by Lema and Mitchell, the AMBTC lossy compression technique [2] is a variant of BTC, in which image blocks are represented by two quantization levels (QLs) and a bitmap. Since the computation cost of AMBTC is even lower than BTC, it has been widely applied in diverse areas, such as remote sensing and portable devices with limited power sources. Therefore, like the authentication methods for VQ and JPEG compressed images [3–6], the investigation of authentication for AMBTC compressed codes has received much attention, and subsequently, a number of authentication methods [7–13] in this field have been proposed in the past few years. These methods adopt spatial-domain data embedding techniques [14–19] or AMBTC-based compressed domain data hiding methods [20–23] to embed authentication codes (ACs) for the purpose of authentication.

In 2013, Hu et al. [7] proposed a joint image coding and authentication method based on AMBTC. In their method, bitmaps are subdivided, and the ACs are embedded into subdivided bitmaps by altering their parity. Wu et al. [8] also proposed an authentication method using similar approaches but embedded ACs with adjustable lengths into bitmaps. Although methods from References [7] and [8] are effective, the modification of bitmaps may cause larger distortions. Another authentication method was proposed in Reference [9] to embed the ACs into QLs rather than the bitmap. To achieve this goal, Reference [9] modifies the QLs such that the computed parity values are equivalent to the generated ACs.

The aforementioned methods [7–9] are early works of the AMBTC image authentication. Recent works [10–13] exploited the merits of the aforementioned methods in References [7–9] and proposed a more efficient authentication scheme with improved image quality. In 2014, Lin et al. [10] proposed an improved ACs generation method for AMBTC authentication. The embedding positions of ACs are generated using a pseudo-random number generator (PRNG), and the ACs are embedded into QLs by referencing the generated embedding positions. In 2016, Zhong et al. [11] employed the exploring modification direction (EMD) technique to embed ACs into one of the two QLs. Since EMD is an efficient embedding method that preserves image fidelity as much as possible, this method provides a better image quality than prior works. In the same year, Li et al. [12] proposed an effective authentication scheme with fine image quality for AMBTC compressed images. The authentication information is firstly converted into digits in a specific base, and these digits are then embedded into the QLs under the guidance of specially designed reference matrices. The image quality obtained by the method in Reference [12] is comparable to that of the method in Reference [11]; however, the bitmap is unprotected in the method in Reference [12]. In 2018, Chen et al. [13] also proposed an efficient method to authenticate the AMBTC codes with fine image quality and improved detectability. While the existing methods in References [10–13] protect the AMBTC codes to a large extent, some security leakage in these methods should not be neglected. Otherwise, they are likely vulnerable to some intentional attacks. For example, the bitmap in Li et al.'s method is unprotected, and thus any modification to the bitmap will not arouse suspicion in the authentication stage.

In this paper, we propose a simple yet efficient authentication method for AMBTC codes. The ACs are generated by hashing the most significant bits (MSBs) of QLs and the bitmap, then the hashed results are folded and embedded into the least significant bits (LSBs) of QLs. We also perturb the MSBs of QLs to ensure that the distortion in the marked image block is the least. The contributions of the proposed method lie in two aspects, namely generating content-dependent ACs and presenting MSBs perturbation technique. Since some tampering would fail the detection due to irrelevance between the AMBTC codes and the ACs, we employ bitmaps and information extracted from two QLs to produce ACs. Therefore, the AMBTC codes can be fully protected and the tampering of various types can be successfully detected. Meanwhile, the MSBs perturbation helps to improve the image quality by finding the optimized ACs with minimized distortion. As a result, the obtained image quality achieves a very satisfactory result.

The rest of this paper is organized as follows. Section 2 describes the AMBTC compression method and some prior works. Section 3 presents the proposed method, while Section 4 gives the experimental results. Discussions and concluding marks are given in the last two sections.

2. Related Works

In this section, the AMBTC compression method will be introduced in short, while the Li et al.'s method, which will be compared with our method, will also be briefly described in this section.

2.1. AMBTC Compression Technique

AMBTC is a lossy compression technique in which image blocks are represented by two QLs and a bitmap. Let O be the original image to be compressed. Firstly, partition O into non-overlapping blocks $O = \{O_i\}_{i=1}^{N}$ of size $m \times m$, where N is the total number of blocks. We denote by $O_{i,j}$ the j-th pixel in O_i; Therefore, $O_i = \{O_{i,j}\}_{j=1}^{m\times m}$. Let v_i be the mean value of $\{O_{i,j}\}_{j=1}^{m\times m}$. To get the bitmap $B_i = \{B_{i,j}\}_{j=1}^{m\times m}$, if $O_{i,j} \leq v_i$, $B_{i,j}$ is set 0, while $O_{i,j} > v_i$, $B_{i,j}$ is set to 1. The lower and upper QLs, denoted by a_i and b_i respectively, can then be obtained by using the following equations

$$a_i = \frac{1}{p}\sum_{B_{i,j}=0} O_{i,j} \tag{1}$$

$$b_i = \frac{1}{m \times m - p}\sum_{B_{i,j}=1} O_{i,j} \tag{2}$$

where p denotes the number of bits valued 0 in B_i. Therefore, the compressed code of O_i can be represented by the trio $\{a_i, b_i, B_i\}$. Each block is processed in the same manner, and the final compressed codes $\{a_i, b_i, B_i\}_{i=1}^{N}$ can be constructed. To decode the i-th image block $\hat{I}_i = \{\hat{I}_{i,j}\}_{j=1}^{m\times m}$ from the i-th trio $\{a_i, b_i, B_i\}$, bits in $B_i = \{B_{i,j}\}_{j=1}^{m\times m}$ are visited. If $B_{i,j} = 0$, $\hat{I}_{i,j} = a_i$ is set; otherwise, $\hat{I}_{i,j} = b_i$ is set. Repeat the same decoding procedures and the final AMBTC decompressed image can be obtained.

Take the following as an example. Let the original image block be O_i = [100 110 137 142; 86 124 142 150; 97 62 63 112; 93 84 94 88], where a semicolon represents the end of one row. Averaging the block, v_i can be obtained and rounded as 105. Therefore, we have B_i = [0111; 0111; 0001; 0000]. Using Equation (1), we have a_i= 85; likewise, we have b_i= 131 by using Equation (2). In a word, the compressed trio of O_i is $\{a_i, b_i, B_i\}$ = {85, 131, [0111; 0111; 0001; 0000]}. To decode the compressed trio, 85 and 131 are used to replace 0 and 1 respectively, and thus the decoded block should be $\hat{I}_i$ = [85 131 131 131; 85 131 131 131; 85 85 85 131; 85 85 85 85].

2.2. Lin et al.'s Method

Lin et al.'s AMBTC authentication method [10] embeds authentication codes ac of length $|ac|$ into a pair of QLs. To ensure the image fidelity, their method uses the authentication index (denoted by ai, $ai \in [0, 2^{|ac|} - 1]$) and position index (denoted by pi, $pi \in [0, 7]$) to generate the position codes, which are used as the guides for ACs embedment. The authentication index ai and position index pi are generated by a PRNG. Based on pi and $|ac|$, the position code, denoted by pc, is generated. The k-th bit of ac, denoted by ac_k, is then generated by

$$ac_k = (ai_k + pv_k)\mathrm{mod}2 \tag{3}$$

where pv_k is the parity of k-th subdivided bitmap, and ai_k is the k-th bit of ai. Finally, ac is embedded into the bits of QLs according to a set of predefined rules.

The authentication procedure is quite simple and straightforward. The embedded ACs are extracted and are compared with the generated ACs. If they are identical, the block is judged as an untampered block. Otherwise, the block is judged as a tampered one. The detailed embedding and authentication procedures can be found in Reference [10].

2.3. Zhong et al.'s Method

Zhong et al. [11] in 2016 proposed a high-fidelity authentication method by employing the EMD embedding technique to embed ACs into one of the two QLs. To embed the ACs into the i-th AMBTC block, the authentication code ac_i, which consists of a digit of base 7, is generated using the equation

$$ac_i = rv_i \mathrm{mod} 7 \tag{4}$$

where rv_i is a random value generated by using a PRNG. A parity value pv_i is generated by using the equation

$$pv_i = (a_i + 2 \times b_i + 3S_{B_i}) \mathrm{mod} 7 \tag{5}$$

where a_i, b_i, and S_{B_i} are the lower quantization level, upper quantization level, and the summation of the bitmap B_i, respectively. If the calculated pv_i is equal to ac_i, no change to the QLs are required. Otherwise, a_i and b_i should be adjusted such that pv_i equals ac_i. The detailed adjustment processes can be found in Reference [11].

2.4. Li et al.'s and Chen et al.'s Methods

In 2016, Li et al. [12] proposed an efficient authentication method for AMBTC compressed codes with higher security and fine image quality. It adopts a user-specified reference matrix R_λ to embed an authentication digit $ac_{i,\lambda}$ in the base λ, into the two QLs of the i-th block. $ac_{i,\lambda}$ is generated by using the equation

$$ac_{i,\lambda} = rv_i \mathrm{mod} \lambda \tag{6}$$

where rv_i is a random value generated from a PRNG. The value of λ can be set to 3, 4, or 5, and the corresponding reference matrices are shown in Figure 1. To embed the authentication information, the ACs are generated from a random seed and are converted into digits in the base λ. By choosing a different λ and the number of digits, the size of the authentication information can be adjusted according to the requirement.

Without loss of generality, we assume two authentication digits $ac^1_{i,\lambda}$ and $ac^2_{i,\lambda}$ in the base λ are to be embedded into two QLs a_i and b_i. The embedment is conducted by solving a'_i and b'_i in the following optimization problem:

$$\text{Minimize}: \quad (a'_i - a_i)^2 + (b'_i - b_i)^2 \tag{7}$$

$$\text{Subject to}: \quad R_\lambda(a'_i \mathrm{mod} \lambda, b'_i \mathrm{mod} \lambda) = ac^1_{i,\lambda} \tag{8}$$

$$R_\lambda(a'_i \mathrm{mod} \lambda, (b'_i + 1) \mathrm{mod} \lambda) = ac^2_{i,\lambda} \tag{9}$$

$$|a'_i - a_i| \le \lambda \tag{10}$$

$$|b'_i - b_i| \le \lambda \tag{11}$$

Let $\hat{a}_i$ and $\hat{b}_i$ be the solution to the above optimization problem. The marked AMBTC trio can thus be obtained by $\{\hat{a}_i, \hat{b}_i, \hat{B}_i\}$, where $\hat{B}_i = B_i$.

Chen et al. [13] recognize that Li et al.'s method only protects the QLs of an AMBTC block but ignores the bitmap. To solve this problem, they take the bitmap into account and modify the rule of generation for the i-th ACs using the following equation

$$ac_i = (rv_i \oplus B_i)\text{mod}\lambda \quad (12)$$

where $\oplus$ is the exclusive-OR operator. The only difference between Chen et al. and Li et al.'s methods is the generation of ACs. The ACs embedment and the authentication procedures of these two methods are the same.

	0	**1**	**2**
0	0	1	0
1	1	2	1
	2	2	0

(**a**)

	0	**1**	**2**	**3**
0	0	0	1	1
1	0	2	1	3
2	0	3	3	2
3	1	2	2	3

(**b**)

	0	**1**	**2**	**3**	**4**
0	0	1	1	4	0
1	1	0	2	3	4
2	1	2	0	3	3
3	2	1	3	0	4
4	2	4	4	3	2

(**c**)

Figure 1. Reference matrices used in Li et al.'s method. (**a**) $\lambda = 3$; (**b**) $\lambda = 4$; (**c**) $\lambda = 5$.

3. Proposed Method

For an image authentication method, the protection of image integrity is always the primary concern. However, the authentication information generated by some of the existing methods is independent of to-be-protected AMBTC codes. As a result, a subtle modification to the marked AMBTC codes will not affect the embedded information. Therefore, a simple tamper can easily elude the detection of those methods. For example, Hu et al. [7] and Wu et al. [9] embed the ACs by using the odd/even parity of a bitmap. However, flipping two bits in a bitmap at the same time does not change the parity of the bitmap, and thus, these two methods fail to detect the tampering of this type. In Reference [10], the position codes are generated using the four MSBs of the two QLs. The ACs are then embedded into the first, second, and/or third LSBs, according to the generated position codes. Obviously, the fourth LSB in Reference [10] is unprotected and any modification to this bit will not arouse the detection using the method in Reference [10]. Similarly, Reference [12] uses Equations (8) and (9) to detect whether the marked AMBTC codes $\{\hat{a}_i, \hat{b}_i, \hat{B}_i\}$ are tampered or not. However, since $(\hat{a}_i \pm k\lambda)\text{mod}\lambda = \hat{a}_i\text{mod}\lambda$ and $(\hat{b}_i \pm k\lambda)\text{mod}\lambda = \hat{b}_i\text{mod}\lambda$, it is easy to see that the tampering by adding or subtracting $\hat{a}_i$ or $\hat{b}_i$ by $k\lambda$ cannot be detected by the method in Reference [12] for any integer k.

The weak design of the aforementioned methods means that some intentional modifications are undetectable, leading to false negative detections (tampered but judged as untampered) on tampered codes. For example, Figure 2a shows the marked image obtained by the Li et al.'s method, while Figure 2b shows the tampering of marked Lena image by splicing a rose and a daisy onto Lena's hat and shoulder, respectively. The splicing of the rose image is conducted by directly placing the AMBTC codes of the rose image onto Lena's hat. The splicing of daisy image is performed in a similar way; however, the quantization levels of daisy image with AMBTC codes $\{a_i^d, b_i^d, B_i^d\}_{i=1}^{Nt}$, where Nt is the total number of blocks, are subtly adjusted in the following manner. Let $\{\hat{a}_i, \hat{b}_i, \hat{B}_i\}_{i=1}^{Nt}$ be the marked to-be-tampered AMBTC codes. The tampered codes $\{\hat{a}_i^d, \hat{b}_i^d, \hat{B}_i^d\}_{i=1}^{Nt}$ used to replace $\{\hat{a}_i, \hat{b}_i, \hat{B}_i\}_{i=1}^{Nt}$ can be obtained by finding $\hat{a}_i^d$ and $\hat{b}_i^d$ with the smallest $|\hat{a}_i^d - a_i^d| + |\hat{b}_i^d - b_i^d|$ while satisfying the conditions $R_\lambda(\hat{a}_i^d\text{mod}\lambda, \hat{b}_i^d\text{mod}\lambda) = ac_\lambda^1$ and $R_\lambda(\hat{a}_i^d\text{mod}\lambda, (\hat{b}_i^d + 1)\text{mod}\lambda) = ac_\lambda^2$ (Equations (8) and (9)). Figure 2c gives tampered regions, which are denoted by black dots.

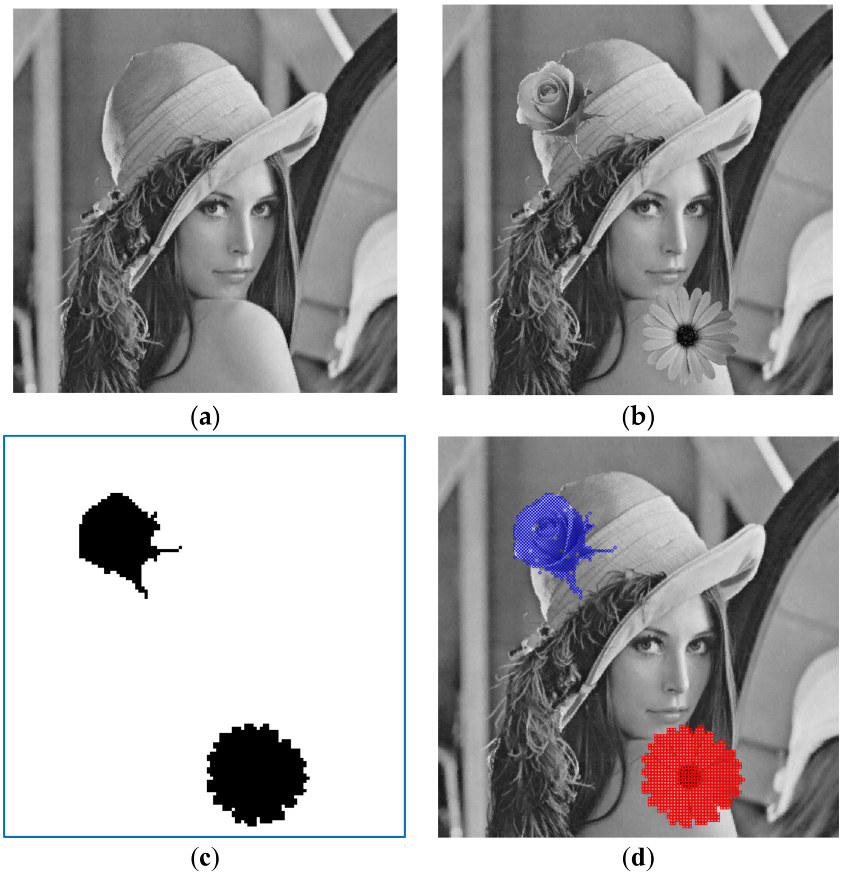


Figure 2. Weakness illustration of Li et al.'s method. (**a**) Marked image of Li et al.'s method; (**b**) Tampered image; (**c**) Tampered regions; (**d**) Detection results, where blue denotes true positive detection, while red denotes false negative detection.

Figure 2d shows the detection result of Li et al.'s method, where the true positive detections (tampered and judged as tampered) are marked by blue circles, whereas the false negative detections are marked by red square marks. As seen in Figure 2d, Li et al.'s method successfully detects the tampering by splicing the rose image. However, splicing the daisy image is totally undetectable in Li et al.'s method. A similar weakness also exists in the methods in References [7,9,10,13], in which some intentional modifications to the marked images cannot be successfully detected by these authentication systems.

In the proposed method, we hash the MSBs of the two QLs and the bitmap, and the hashed results are then folded and embedded into the LSBs of the QLs. In this way, any modification to the marked AMBTC codes will cause an unmatched error in the authentication process. To enhance the image quality, the MSBs are further perturbed such that the resultant marked QLs have the shortest distance to their original ones. The authentication can be performed by comparing the re-generated ACs with the extracted ACs from the to-be-authenticated AMBTC codes. Figure 3 shows the flowchart of the proposed method, which will be detailed in the following sections.

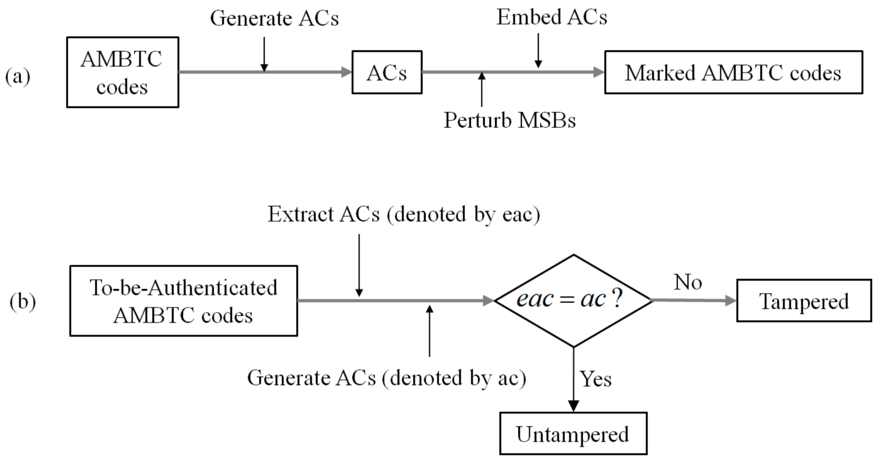


Figure 3. Flowchart of the proposed method. (**a**) Procedures of ACs generation and embedment; and (**b**) authentication procedures.

3.1. Generation of ACs

The proposed method generates the ACs with the MSBs of the lower and upper QLs and the bitmap, and then embeds the ACs into the LSBs of the two QLs. The embedment of ACs using the LSB replacement in the proposed method is termed LSBR for short.

Suppose the QLs are of eight bits. Let α and β be the number of LSBs of a_i and b_i used to embed the ACs, respectively. Let a_i^M be the $8-\alpha$ MSBs of a_i, and $(a_i^M)_d$ be the decimal values of a_i^M. Similarly, we denote by b_i^M the $8-\beta$ MSBs of b_i, and $(b_i^M)_d$ the decimal values of b_i^M. To generate ACs ac_i of the i-th trio $\{a_i, b_i, B_i\}$, we hash $(a_i^M)_d$, $(b_i^M)_d$, and bitmap B_i using a hash function [24]. The hashed result, which is a 128-bit bitstream, is folded to obtain ac_i of $\alpha+\beta$ bits. The folding can be conducted by repeating the bitwise xor operation on two folded sub-bitstreams. The generation of ac_i in the proposed method can be described as a function $f_{\alpha+\beta}$ where the inputs are $(a_i^M)_d$, $(b_i^M)_d$, and B_i, i.e.,

$$ac_i = f_{\alpha+\beta}((a_i^M)_d, (b_i^M)_d, B_i) \tag{13}$$

3.2. Embedment of ACs

The generated ac_i is embedded into a_i and b_i using the LSB replacement, where a_i carries the first α bits, whereas b_i carries the other β bits. We denote by $\hat{a}_i$ and $\hat{b}_i$ the embedded results of a_i and b_i, respectively. Specifically, let $(ac_i^\alpha)_d$ be the decimal value of the first α bit of ac_i, and $(ac_i^\beta)_d$ be the decimal value of the rest of the bits of ac_i. Therefore, the embedment of ac_i into a_i and b_i can be simply calculated using the following two equations:

$$\hat{a}_i = (a_i^M)_d \times 2^\alpha + (ac_i^\alpha)_d \tag{14}$$

$$\hat{b}_i = (b_i^M)_d \times 2^\beta + (ac_i^\beta)_d \tag{15}$$

The final marked AMBTC trio is denoted by $\{\hat{a}_i, \hat{b}_i, \hat{B}_i\}$, where $\hat{B}_i = B_i$. The procedures of generation and embedment of ACs are shown in Figure 4.

Here is an example of the proposed embedding method. Suppose $\{a_i, b_i, B_i\}$= {85, 131, [0111; 0111; 0001; 0000]} be the original trio, and we assume $\alpha = 2$ and $\beta = 1$. Since $a_i^M = 010101_2$ and $b_i^M = 1000001_2$, we have $(a_i^M)_d = 21$ and $(b_i^M)_d = 65$. Suppose $ac_i = f_{2+1}(21, 65, B_i) = 110_2$ and therefore, $ac_i^2 = 11_2$ and $ac_i^1 = 0_2$. According to Equations (14) and (15), we have $\hat{a}_i = 21 \times 2^2 + 3 = 87$ and $\hat{b}_i = 65 \times 2^1 + 0 = 130$.

$$\begin{tikzcd}
 & \text{Original trio: } \{a_i, b_i, B_i\} \arrow[d, shift right=3] \arrow[d] \arrow[d, shift left=3] & \\
 & ac_i = f_{\alpha+\beta}((a_i^M)_d, (b_i^M)_d, B_i) \arrow[dl] \arrow[d] \arrow[dr] & \\
\hat{a}_i = (a_i^M)_d \times 2^{\alpha} + (ac_i^{\alpha})_d \arrow[dr] & \hat{b}_i = (b_i^M)_d \times 2^{\beta} + (ac_i^{\beta})_d \arrow[d] & B_i \arrow[dl] \\
 & \text{Marked trio: } \{\hat{a}_i, \hat{b}_i, \hat{B}_i\} &
\end{tikzcd}$$

Figure 4. Generation and embedment of ACs.

3.3. The MSBs Perturbation Technique

Intuitively, the slight modification of $(a_i^M)_d$ and $(b_i^M)_d$ alters the generated $(\alpha + \beta) -$ bit ACs, which in turn changes the marked QLs a_i' and b_i' (see Equations (14) and (15)). We therefore slightly perturb $(a_i^M)_d$ and $(b_i^M)_d$, then perform the embedding technique described in Section 3.2 to find a pair of marked QLs a_i' and b_i' with the shortest distance to a_i and b_i. In this way, the quality of the reconstructed AMBTC image will be enhanced because the marked QLs have smaller distortions.

Let $(a_i'^M)_d$ and $(b_i'^M)_d$ be the perturbed value of $(a_i^M)_d$ and $(b_i^M)_d$, respectively. The optimal perturbed values can then be calculated by solving the optimization problem:

$$\text{Minimize}: \ (a_i' - a_i)^2 + (b_i' - b_i)^2 \tag{16}$$

$$\text{Subject to}: \ ac_i' = f_{\alpha+\beta}((a_i'^M)_d, (b_i'^M)_d, B_i) \tag{17}$$

$$a_i' = (a_i'^M)_d \times 2^{\alpha} + (ac_i'^{\alpha})_d, \ b_i' = (b_i'^M)_d \times 2^{\beta} + (ac_i'^{\beta})_d \tag{18}$$

$$|(a_i^M)_d - (a_i'^M)_d| \le 1, \ |(b_i^M)_d - (b_i'^M)_d| \le 1 \tag{19}$$

where $(ac_i'^{\alpha})_d$ and $(ac_i'^{\beta})_d$ are the decimal values of the first α bits and the remaining β bits of ac_i', respectively. We find that perturbing $(a_i^M)_d$ and $(b_i^M)_d$ by one unit is enough to obtain the optimal solution. As a result, Equation (19) is added as the constraints to reduce the problem space. Since only nine possible combinations of $(ac_i'^{\alpha})_d$ and $(ac_i'^{\beta})_d$ will be employed in solving the optimization problem, the solutions to this problem can be solved with few calculations. Let $(a_i^{M*})_d$ and $(b_i^{M*})_d$ be the solutions to the optimization problem, and $ac_i^{\alpha*}$ and $ac_i^{\beta*}$ be the corresponding ACs, respectively. The marked trio $\{\hat{a}_i, \hat{b}_i, \hat{B}_i\}$, where $\hat{a}_i = (a_i^{M*})_d \times 2^{\alpha} + (ac_i^{\alpha*})_d$, $\hat{b}_i = (b_i^{M*})_d \times 2^{\beta} + (ac_i^{\beta*})_d$, and $\hat{B}_i = B_i$ can then be obtained. Each trio is processed in the same manner, and we have the final marked trios $\{\hat{a}_i, \hat{b}_i, \hat{B}_i\}_{i=1}^{N}$. We term the proposed MSBs perturbation technique as MSBP, and Figure 5 gives brief procedures for calculating $\hat{a}_i$ and $\hat{b}_i$.

Figure 6 gives the comparison of image quality improvement when using the MSBP technique. Figure 6a shows the AMBTC compressed images, whereas Figure 6b,c show the marked image using the proposed LSBR and the MSBP techniques with $\alpha = \beta = 4$, respectively. With the MSBP technique, blocks had smaller than or equal distortions to those when LSBR technique is applied. Those blocks with smaller distortions are marked by black circles, as shown in Figure 6d. As seen in Figure 6, when compared to the original AMBTC compressed image (Figure 6a), LSBR distorts the image significantly, as the contours can obviously be seen on Lena's shoulder (Figure 6b). On the other hand, the image quality obtained by MSBP offered better image quality (See Figure 6c) than those of the LSBR method, as the distortions were less apparent. The improvements can be shown in Figure 6d, where the black circles were densely distributed over the image, indicating that the MSBP technique indeed helped the enhancement of image quality.

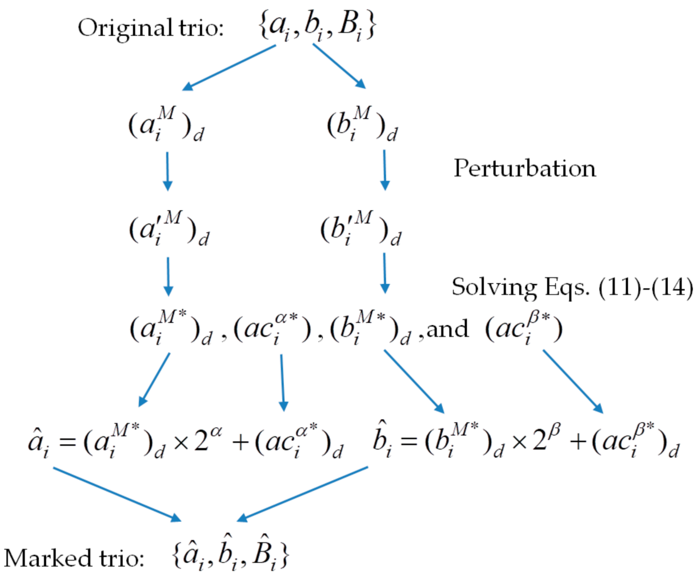

Original trio: $\{a_i, b_i, B_i\}$
$(a_i^M)_d$
$(b_i^M)_d$
Perturbation
$(a_i'^M)_d$
$(b_i'^M)_d$
Solving Eqs. (11)-(14)
$(a_i^{M*})_d$, $(ac_i^{\alpha*})$, $(b_i^{M*})_d$, and $(ac_i^{\beta*})$
$\hat{a}_i = (a_i^{M*})_d \times 2^\alpha + (ac_i^{\alpha*})_d$
$\hat{b}_i = (b_i^{M*})_d \times 2^\beta + (ac_i^{\beta*})_d$
Marked trio: $\{\hat{a}_i, \hat{b}_i, \hat{B}_i\}$

Figure 5. MSBs perturbation technique.

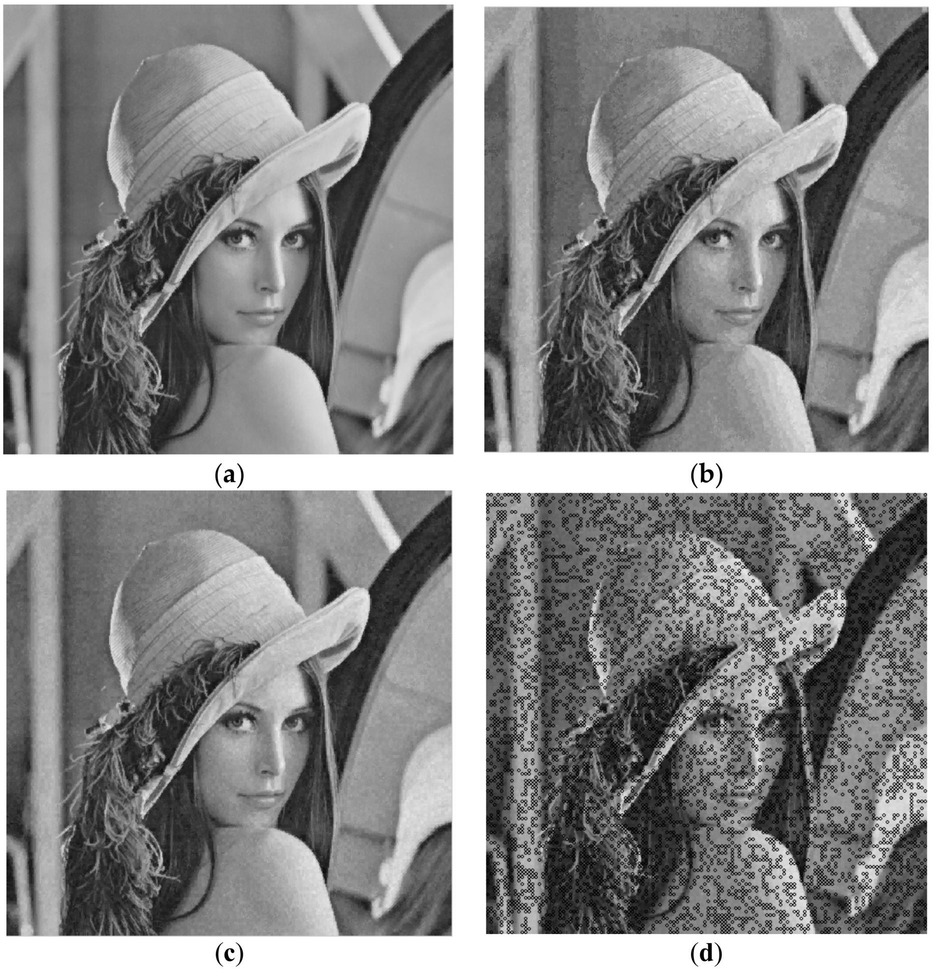

(**a**) (**b**)
(**c**) (**d**)

Figure 6. Comparison of image quality at $|ac| = 8$. (**a**) AMBTC compressed image, PSNR = 33.24 dB; (**b**) LSBR, PSNR = 29.40 dB; (**c**) MSBP, PSNR = 30.41 dB; (**d**) Improved blocks.

We continue the example given in Section 3.2 to illustrate the proposed MSBP technique. Since $(a_i^M)_d = 21$ and $(b_i^M)_d = 65$, the nine possible pairs of $((a_i'^M)_d, (b_i'^M)_d)$ are (20,64), (21,64), (22,64), (20,65), (21,65), (22,65), (20,66), (21,66), and (22,66). Suppose the obtained ac_i' using these perturbed values are 100_2, 110_2, 101_2, 000_2, 110_2, 000_2, 010_2, 010_2, and 100_2, respectively. By applying Equation (18), the perturbed pairs (a_i', b_i') are (82,128) (87,128), (90,129), (80,130), (87,130), (88,130), (81,132), (85,132), (90,132), and the sum of the squared distance (Equation (16)) for these nine perturbed pairs are 18, 13, 29, 26, 5, 10, 17, 1, and 26, respectively. Because the eighth pair gives the smallest distance, we have $(a_i^{M*})_d = 21$, $(ac_i^{2*})_d = 1$, $(b_i^{M*})_d = 66$, and $(ac_i^{1*})_d = 0$. Therefore, $\hat{a}_i = 85$ and $\hat{b}_i = 132$ (see Equation (18)) are the solutions to this problem, and thus we have the marked trio $\{\hat{a}_i, \hat{b}_i, \hat{B}_i\}$ = {85, 132, [0111; 0111; 0001; 0000]}.

3.4. The Authentication Procedures

The proposed method adopts two stages to detect whether the AMBTC codes are tampered or not. Let $\{\widetilde{a}_i, \widetilde{b}_i, \widetilde{B}_i\}_{i=1}^N$ be the AMBTC codes to be authenticated. To authenticate the i-th trio $\{\widetilde{a}_i, \widetilde{b}_i, \widetilde{B}_i\}$ in the first stage, the decimal values $(\widetilde{a}_i^M)_d$ of the $8 - \alpha$ MSBs of $\widetilde{a}_i$, the decimal values $(\widetilde{b}_i^M)_d$ of the $8 - \beta$ MSBs of $\widetilde{b}_i$, and the bitmap $\widetilde{B}_i$ are hashed and folded to generated ACs $a\widetilde{c}_i$ of $\alpha + \beta$ bits. Then, the α LSBs of $\widetilde{a}_i$ and the β LSBs of $\widetilde{b}_i$ are concatenated to form $(\alpha + \beta) -$ bit extracted codes $e\widetilde{c}_i$. If $a\widetilde{c}_i$ and $e\widetilde{c}_i$ are identical, $\{\widetilde{a}_i, \widetilde{b}_i, \widetilde{B}_i\}$ is judged as an untampered trio. Otherwise, the trio has been tampered. The proposed method also adopts a second stage authentication to refine the detection results, as used in Li et al.'s method. Since the tampered trios are likely to be contiguous, a trio with two adjacent tampered trios is more likely tampered as well. Therefore, if the upper and lower, left and right, upper left and lower right, or upper right and lower left of a trio are tampered, the current trio is also judged as a tampered one. All the trios are detected using the same manner and the final detection result can be obtained.

4. Experimental Results

In this section, we performed several experiments to demonstrate the performance of the proposed method. A total of eight test images obtained from SIPI image database [25] were used to evaluate the applicability of the proposed method, as shown in Figure 7. We also compared the results with some state-of-the-art works, including Lin et al.'s [10], Zhong et al.'s [11], Li et al.'s [12], and Chen et al.'s [13] methods. In all experiments, a block of size 4 × 4 was used.

The peak signal-to-noise ratio (PSNR) metric was used to measure the image quality. A higher PSNR indicated that the measured image offered a closer image quality to its original one. The PSNR is defined by

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\frac{1}{N_p}\sum_{i=1}^{N_p}(x_i - x_i')^2} \tag{20}$$

where x_i and x_i' are the pixel values of the original image and the marked image, respectively, and N_p is the total number of pixels.

4.1. Performance Evaluation of the Proposed Method

Table 1 gives the PSNRs of the marked images of the proposed method under different lengths of ACs, which is denoted by $|ac|$ ($|ac| = \alpha + \beta$ in the proposed method). The term LSBR indicates the proposed LSB replacement technique is used for embedding the ACs, as described in Section 3.2, whereas the term MSBP represents the most significant bit perturbation technique is applied. We also list the PSNRs of the unembedded AMBTC compression images (labeled by AMBTC) for the purpose of comparison.

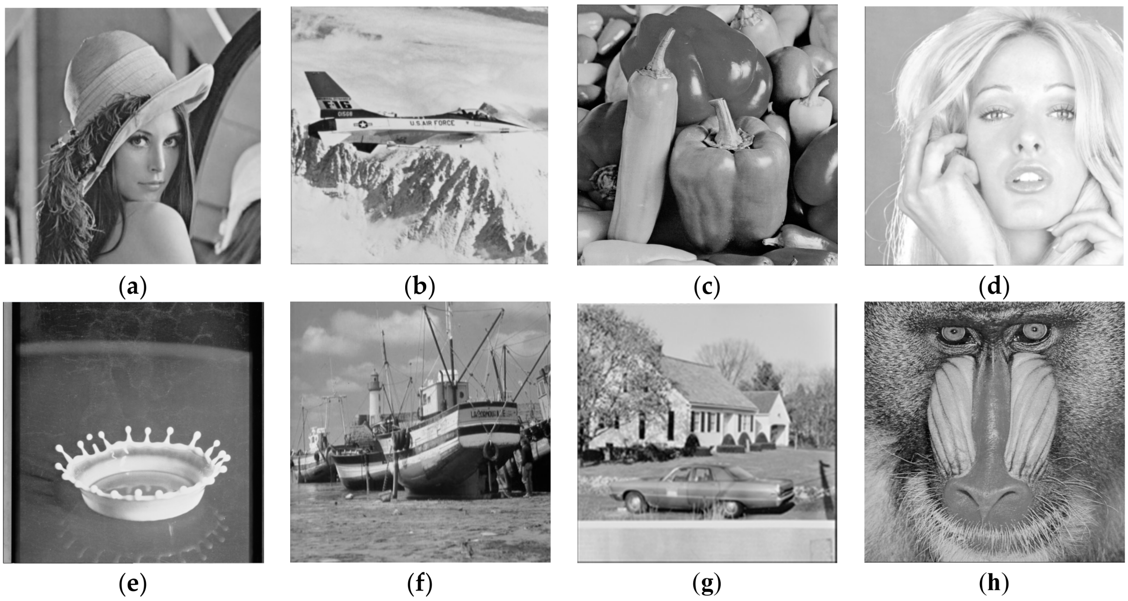


Figure 7. Eight test images. (**a**) Lena; (**b**) Jet; (**c**) Peppers; (**d**) Tiffany; (**e**) Splash; (**f**) Boat; (**g**) House; (**h**) Baboon.

Table 1. Image qualities in dB for various lengths of $\lvert ac\rvert = \alpha + \beta$.

$\lvert ac\rvert$	**Method**	**Lena**	**Jet**	**Peppers**	**Tiffany**	**Splash**	**Boat**	**House**	**Baboon**
	AMBTC	33.23	31.97	33.42	35.76	36.81	31.16	30.89	26.98
$\alpha = 1$	LSBR	33.16	31.92	33.35	35.64	36.65	31.12	30.85	26.96
$\beta = 1$	MSBP	33.17	31.92	33.36	35.66	36.67	31.12	30.85	26.96
$\alpha = 2$	LSBR	32.89	31.71	33.06	35.18	36.07	30.95	30.68	26.89
$\beta = 2$	MSBP	33.00	31.79	33.17	35.34	36.28	31.01	30.74	26.92
$\alpha = 3$	LSBR	31.96	31.00	32.10	33.69	34.33	30.34	30.08	26.64
$\beta = 3$	MSBP	32.35	31.97	32.50	34.30	35.02	30.60	30.35	26.75
$\alpha = 4$	LSBR	29.40	28.91	29.55	30.31	30.60	28.50	28.34	25.74
$\beta = 4$	MSBP	30.41	29.75	30.54	31.58	31.98	29.23	29.10	26.13

Table 1 shows that the PSNR of the Baboon image is the lowest while the Splash image is the highest for every $\lvert ac\rvert$. This was because the fluctuations of pixel values in complex blocks were larger than those of smooth ones, and larger fluctuations were more difficult to be represented by two QLs and a bitmap. Therefore, complex images have a lower PSNR than those of smooth ones.

Note that the proposed MSBP was an improved version of LSBR. The improvement was minor when $\lvert ac\rvert$ is small; however, the improvement became significant as $\lvert ac\rvert$ becomes large. For example, the improvement of the Lena image at $\lvert ac\rvert = \alpha + \beta = 1 + 1 = 2$ was $33.17 - 33.16 = 0.01$ dB, whereas the improvement was $30.41 - 29.40 = 1.01$ dB when $\lvert ac\rvert = 4 + 4 = 8$. This was because larger $\lvert ac\rvert$ often caused larger distortions, and the proposed MSBP technique effectively reduced those larger distortions to smaller ones.

To demonstrate the detectability of the proposed method, we specifically tampered the Lena image by adding a rose to Lena's hat, as shown in Figure 8a,b, which shows the corresponding tampered regions in black.

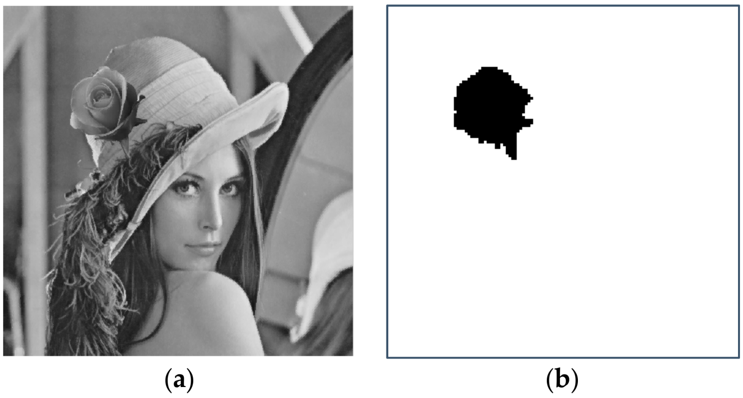


Figure 8. (**a**) Tampered image; and (**b**) tampered regions.

Figure 9 shows the detection result of the first and second stage detections by using $|ac| = 2$ and $|ac| = 4$, respectively. As seen from this figure, $|ac| = 4$ detected better than that of $|ac| = 2$. Moreover, the second stage detection improved the detectability in that some of the undetected tampered blocks in the first stage were now detectable at the second stage. The incorrect detections of the first and second stages are shown in Figure 9c,f, respectively. It was interesting to note that the incorrectly detected blocks were sparsely distributed when only the first stage detection was applied. Nevertheless, only four incorrectly detected blocks appeared at the border of the tampered regions as the second detection was applied.

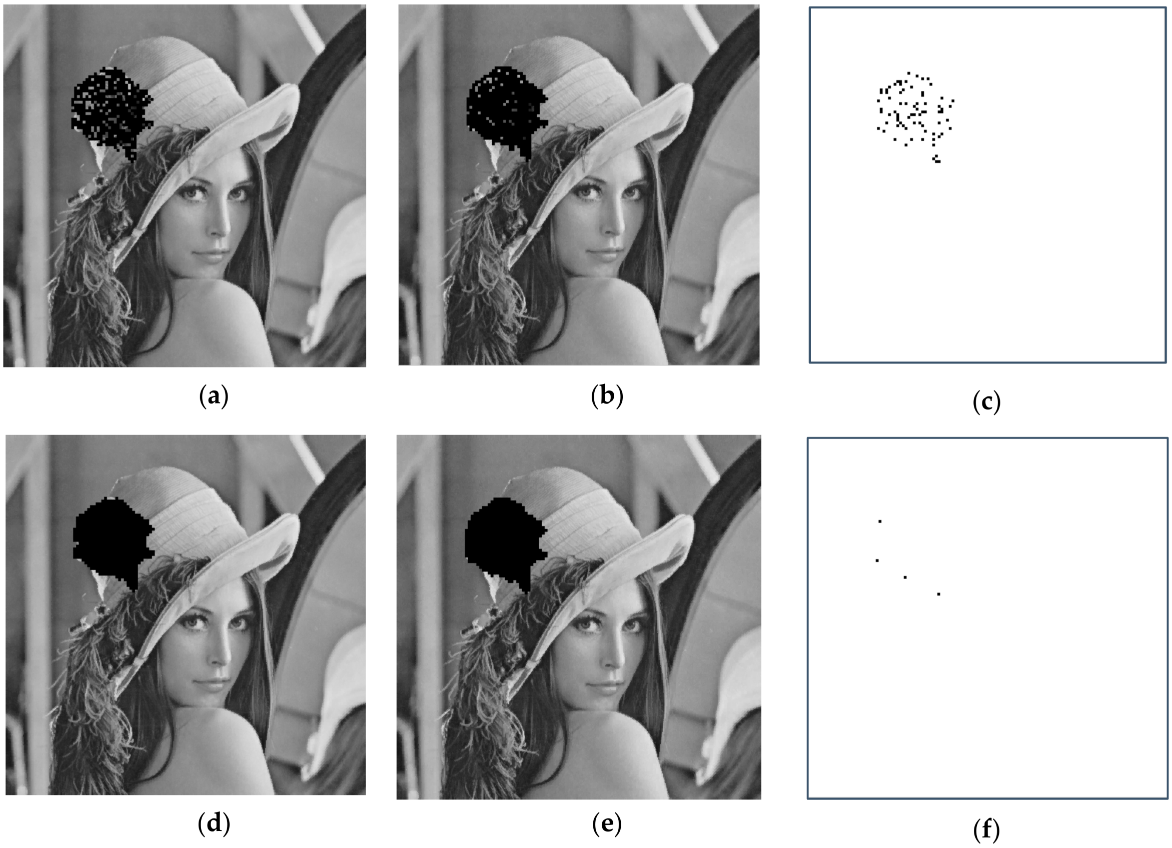


Figure 9. Results of the first and second stage detections. (**a**) 1st stage, $|ac| = 2$; (**b**) 1st stage, $|ac| = 4$; (**c**) 1st stage incorrect detection ($|ac| = 4$); (**d**) 2nd stage, $|ac| = 2$; (**e**) 2nd stage, $|ac| = 4$; (**f**) 2nd stage incorrect detection ($|ac| = 4$).

4.2. Image Quality Comparison with Other Works

In this section, we compare the performance of the proposed method with Lin et al.'s [10], Zhong et al.'s [11], Li et al.'s [12] and Chen et al.'s [13] methods in terms of image quality. To make a fair comparison, some parameters were set to ensure each method achieved its best performance. In Lin et al.'s and the proposed methods, we embed 1 bit into the lower QL if $|ac| = 1$, and embed 1 bit each into the lower and upper QLs if $|ac| = 2$. If $|ac| = 3$, the lower and upper QLs carried one and two bits, respectively. If $|ac| = 4$, each quantization level carried two-bit ACs. In Li et al.'s method, a proper base was selected such that the best image quality was achieved for a given $|ac|$ and λ. In Zhong et al.'s method, authentication digits in base 7 were randomly generated and were embedded into the QLs. The comparison of image quality under various $|ac|$ are shown in Table 2, where the best image quality of test images with the same $|ac|$are highlighted in bold font. Note that since the length of ACs in Zhong et al.'s method was un-adjustable, we did not compare the image quality of Zhong et al.'s method in this table.

Table 2. Comparisons with other works.

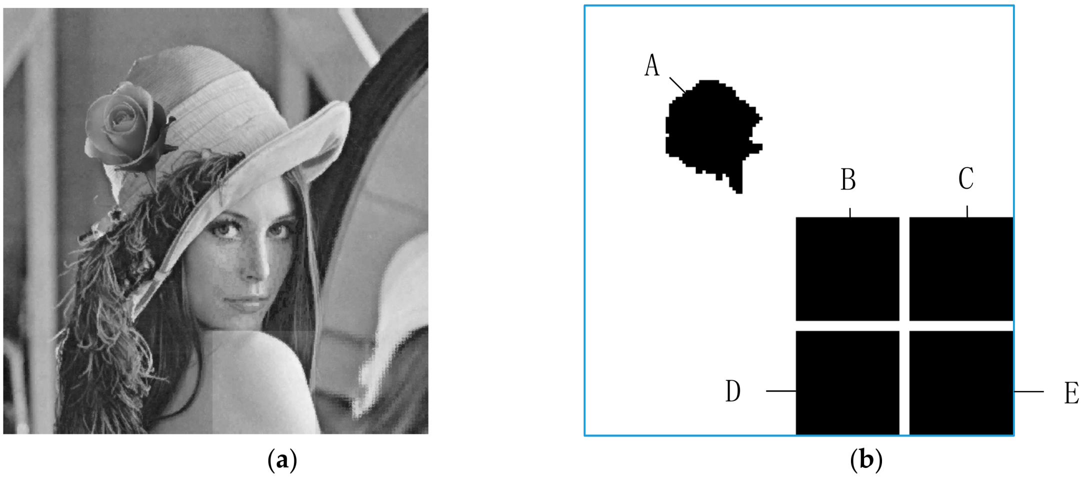
Image		Lena	Jet	Peppers	Tiffany	Splash	Boat	House	Baboon
AMBTC		33.23	31.97	33.42	35.76	36.81	31.16	30.89	26.98
$\|ac\| = 1$	MSBP	**33.20**	**31.94**	**33.38**	**35.70**	**36.73**	**31.14**	**30.87**	26.97
	Lin et al.	33.15	31.89	33.32	35.62	36.62	31.09	30.82	26.96
	Li et al.	33.16	31.93	33.37	35.68	36.70	31.13	30.86	26.97
	Chen et al.	33.16	31.93	33.36	35.68	36.70	31.12	30.85	**26.98**
$\|ac\| = 2$	MSBP	**33.16**	**31.92**	**33.35**	35.63	36.65	**31.12**	**30.85**	**26.96**
	Lin et al.	33.11	31.86	33.28	35.56	36.55	31.07	30.80	26.95
	Li et al.	33.15	31.91	33.34	**35.64**	**36.66**	31.11	30.84	26.95
	Chen et al.	33.15	31.91	33.34	35.63	**36.66**	31.11	**30.85**	26.95
$\|ac\| = 3$	MSBP	33.10	31.87	33.29	35.54	36.53	31.08	30.81	26.94
	Lin et al.	32.74	31.62	32.88	35.09	35.72	30.82	30.58	26.86
	Li et al.	**33.14**	**31.90**	**33.32**	**35.60**	36.60	**31.10**	**30.83**	**26.95**
	Chen et al.	**33.14**	**31.90**	**33.32**	**35.60**	**36.61**	31.09	**30.83**	26.94
$\|ac\| = 4$	MSBP	32.99	31.79	33.17	35.34	36.28	31.01	30.74	26.92
	Lin et al.	32.66	31.55	32.80	34.94	35.57	30.77	30.53	26.84
	Li et al.	**33.03**	**31.82**	**33.21**	**35.41**	**36.36**	**31.03**	**30.77**	**26.93**
	Chen et al.	**33.03**	**31.82**	**33.21**	**35.41**	**36.36**	**31.03**	**30.77**	26.92

Table 2 shows that the image qualities of the proposed method and Li et al.'s method were comparable, whereas the PSNR of Lin et al.'s method was the lowest, especially when $|ac| = 3$ and $|ac| = 4$. This was because Lin et al.'s method could modify the three LSBs of the QLs in these cases, which would lead to more distortions. However, the proposed method only modified the first and the second LSBs. Therefore, the proposed method offered better image quality than that of Lin et al.'s work. Regarding Li et al.'s method, it adopted a reference matrix for data embedment, and thus, had a better embedding performance with larger $|ac|$. With the proposed MSBP technique, the image quality of the proposed method was slightly smaller than or comparable to Li et al.'s work. The image qualities of Zhong et al.'s method were 33.14, 31.88, 33.31, 35.58, 36.63, 31.09, 30.82, and 26.96 for the eight test images when the authentication digit was of base 7, which was equivalent to $|ac| = \log_2 5 \simeq 2.322$ bits. As a result, the amount of information carried by each trio in Zhong et al.'s method was between two to three bits. As seen from the table, the image quality of Zhong et al.'s method was also comparable to that of the proposed and Li et al.'s methods.

4.3. Detectability Comparison with Other Works

With the ease of digital image manipulation, image tampering has become a critical issue in many applications. Tamper detectability has thus become increasingly important for an authentication method. To compare the detectability of Lin et al.'s, Li et al.'s, Zhong et al.'s, and the proposed methods, we applied various attacks on marked AMBTC codes obtained by each method. For all the compared methods, $|ac| = 4$ was used. The tampered image is shown in Figure 10a, while the tampered regions are illustrated in Figure 10b. In tampered region A, we added a rose to Lena's hair. In this region, the corresponding QLs and bitmaps were all modified. In tampered region B, we flipped the fourth LSB of lower QLs. In tampered region C, we randomly flipped a zero-valued bit and a one-valued bit in each bitmap simultaneously. Note that this modification did not alter the parity of the bitmap. We increased the brightness of tampered region D by adding 32 to both the lower and upper QLs. In region E, the bitmap was tampered with by replacing the bitmap with random bits.

Figure 11 shows the detection results, and the results revealed that all methods were capable of detecting the tampering by adding a rose. However, the detection capabilities for some special tampering were varied.



(**a**) (**b**)

Figure 10. (**a**) Tampered image; and (**b**) tampered regions.

Lin et al.'s method successfully detected tampered region C, but totally failed to detect tampered region B (see Figure 10a). The failure was such because the position codes were generated from the four MSBs of the QLs and then embedded into the three LSBs of QLs. As a result, the fourth LSB of QLs was left unprotected. As for Li et al.'s method, while the authentication was conducted by checking whether Equations (8) and (9) were satisfied, the generation of ACs in their method was independent of bitmaps. As a result, any modification to bitmaps (tampered regions C and E) could not be detected by this method. Moreover, because $\lambda = 4$ was set in this experiment, flipping the fourth bit of QLs was also divisible by 4. Besides, the constant 32 is also divisible by 4, and Equations (8) and (9) show that $\hat{a}_i \text{mod} 4 = (\hat{a}_i + 32) \text{mod} 4$ and $\hat{b}_i \text{mod} 4 = (\hat{b}_i + 32) \text{mod} 4$. Therefore, tampered regions B and D could not be detected by Li et al.'s method neither, as shown in Figure 10b. Although Zhong et al.'s method successfully detected tampered regions B, D, and E, it failed to detect tampered region C (see Figure 10c). This was because Zhong et al.'s method used the parity of bitmaps for authentication, yet simultaneously flipping a zero-valued bit and a one-valued bit did not change the parity of bitmaps. Therefore, the tampered region C could not be detected in their method. On the other hand, the proposed method hashed the bitmaps and the MSBs of the QLs, and the hashed results were processed and embedded into the LSBs of QLs. Since any modification to bitmaps and the MSBs of QLs would affect the hashed result, the proposed method successfully detected all the tampered regions (see Figure 10d).

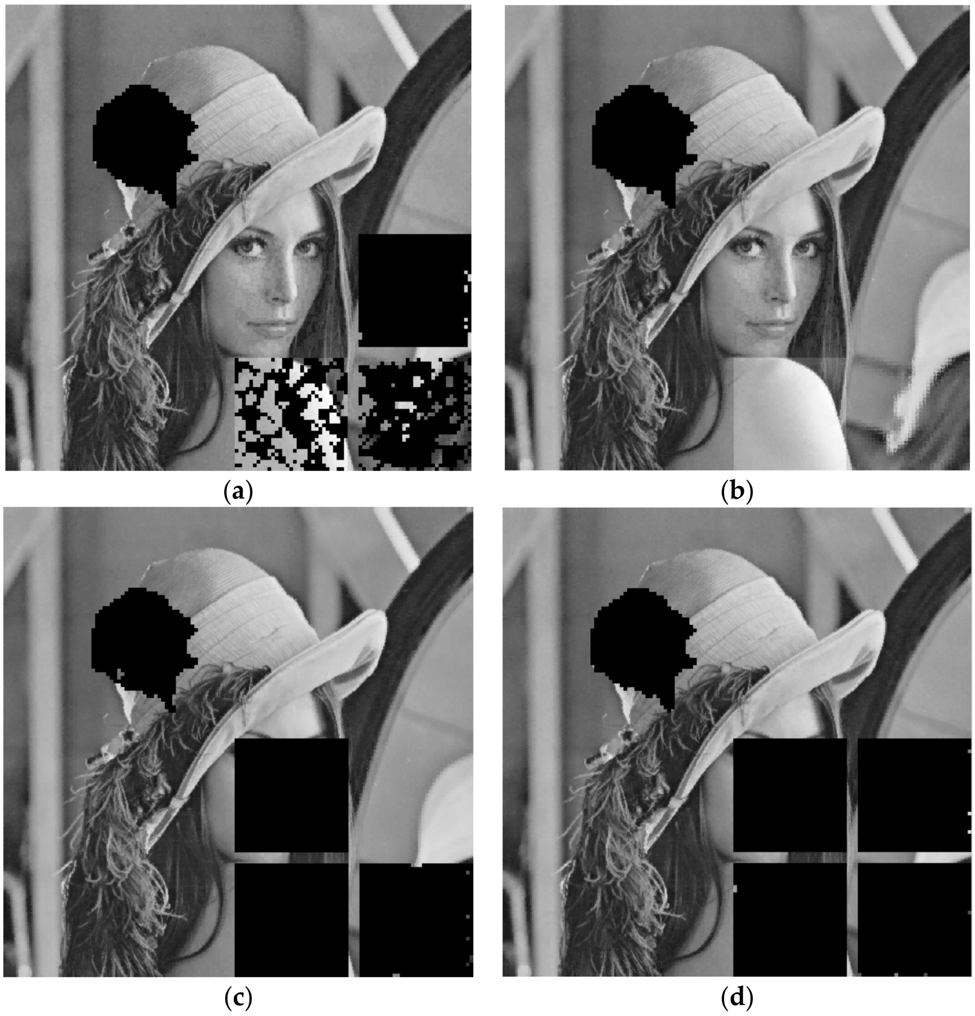


Figure 11. Second stage detection results of each method. (**a**) Lin et al.'s method; (**b**) Li et al.'s method; (**c**) Zhong et al.'s method; (**d**) Proposed method.

In the experiments of this section, the total number of tampered blocks, denoted by NTB, in regions A, B, C, D, and E was 4476. We use TP and FN to denote the number of blocks with true positive detections (tampered and judged as tampered) and false negative detections (tampered but judged as untampered), respectively. The rates of true positive detections (TPR = TP/NTB) and false negative detections (FNR = FN/NTB) of each method are listed in Table 3. A well-designed authentication method should increase the TPR and reduce the FNR as much as possible. While all the compared methods were capable of detecting the splicing tampering (Region A), the proposed method even better detected various types of intentional tampering, since it offered the highest TPR with the lowest FNR.

Table 3. Comparisons of rates of true positive and false negative detections.

Methods	Chen et al.	Li et al.	Lin et al.	Zhong et al.	Proposed
TPR	38.38%	14.10%	60.81%	78.08%	99.66%
FNR	61.62%	85.90%	39.19%	21.92%	0.34%

5. Discussions

The proposed method has been proved effective in detecting diverse tampering while providing a satisfactory image quality. This study proposed LSBR and MSBP techniques, using LSBR to create rooms for embedding ACs in order to identify tampering that escaped the detection of prior methods. Detectability comparisons of the proposed method and the other four related works are given in Table 4. As seen from this table, the compared methods might not always successfully detect various types of tampering. In contrast, the proposed method was capable of detecting all the tampering, indicating that the proposed method indeed outperformed other works in terms of detectability.

Moreover, in general, the marked image quality decreased as the length of the embedded ACs increased. That is, the large length of ACs decreased the image quality but increased the detectability because more authentication information could be embedded into the quantization levels. In contrast, the small length of ACs preserved the image fidelity but reduced the detectability. As a result, the determination of the length of ACs was a trade-off between image quality and detectability. For most of the cases, setting $|ac| = 4$ ($\alpha = 2$ and $\beta = 2$) achieved satisfactory results, both for image quality and detectability. Nevertheless, with the proposed MSBP technique, the marked image quality was always the best for a given length of ACs. The challenge faced in this study was to analyze, debug, and compare the research results in order to avoid already known pitfalls, but there still remains limitations for further improvement. Since the proposed method adopted the LSB replacement technique to embed ACs, it failed to recover the tampered blocks. Future studies may investigate the reversible authentication scheme that has the capability to fully or partially restore the original AMBTC blocks from their tampered ones.

Table 4. Detectability comparisons.

Type of Tampering	**Methods**				
	Lin et al.	**Zhong et al.**	**Li et al.**	**Chen et al.**	**Proposed**
Flip two bits of a bitmap	Yes	No	No	Partially	Yes
Add a constant 32 to QLs	Partially	Yes	No	No	Yes
Flip the fourth bits	No	Yes	No	No	Yes
Replace the bitmap with random bits	Partially	Yes	No	Yes	Yes
Subtract a constant 7 from QLs	Yes	No	Yes	No	Yes
Scramble the bitmap	Partially	No	No	Yes	Yes

6. Conclusions

In this paper, we proposed a novel authentication method to protect the AMBTC compressed codes. The proposed LSBR technique generates the ACs from the MSBs of QLs and the bitmap, and the generated ACs are embedded into the LSBs of the QLs. The proposed MSBP technique perturbs the MSBs of QLs and finds a perturbed MSB that minimizes the distortion when embedded with the authentication codes. The experimental results reveal that the proposed method not only offers a comparable marked image quality but also provides a better detectability than prior state-of-the-art works.

Author Contributions: W.H. conceived the overall ideas of this article. X.Z. proposed some ideas for the proposed detection technique and analyzed the data. D.-C.L. proofread the manuscript and authenticated the data set. X.H. wrote the paper, and C.P. carried out the experiments.

Funding: This work was supported partially by the Ministry of Science and Technology of Taiwan and Chang Gung Memorial Hospital under the grants MOST 107-2221-E-182-014- and BMRPB30, respectively.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Delp, E.; Mitchell, O. Image compression using block truncation coding. *IEEE Trans. Commun.* **1979**, *27*, 1335–1342. [CrossRef]
2. Lema, M.; Mitchell, O. Absolute moment block truncation coding and its application to color image. *IEEE Trans. Commun.* **1984**, *32*, 1148–1157. [CrossRef]

3. Wu, W.C. Quantization-based image authentication scheme using QR error correction. *EURASIP J. Image Video Process.* **2017**, *2017*, 13. [CrossRef]
4. Lee, C.F.; Chen, K.N.; Chang, C.C.; Tsai, M.C. A hierarchical fragile watermarking with vq index recovery. *J. Multimedia* **2011**, *6*, 277–284. [CrossRef]
5. Chen, J.; Chen, T.S.; Cheng, C.Y. New image tampering detection and recovery system of jpeg2000 region of interest area. *Imaging Sci. J.* **2005**, *53*, 12–19. [CrossRef]
6. Tsai, P.Y.; Hu, Y.C.; Chang, C.C. A novel image authentication scheme based on quadtree segmentation. *Imaging Sci. J.* **2005**, *53*, 149–162. [CrossRef]
7. Hu, Y.C.; Lo, C.C.; Chen, W.L.; Wen, C.H. Joint image coding and image authentication based on absolute moment block truncation coding. *J. Electron. Imaging* **2013**, *22*, 013012. [CrossRef]
8. Hu, Y.C.; Lo, C.C.; Wu, C.M.; Chen, W.L.; Wen, C.H. Probability-based tamper detection scheme for BTC-compressed images based on quantization levels modification. *Int. J. Secur. Appl.* **2013**, *7*, 11–32.
9. Wu, C.M.; Hu, Y.C.; Liu, K.Y.; Chuang, J.C. A novel active image authentication scheme for block truncation coding. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2014**, *7*, 13–26.
10. Lin, C.C.; Huang, Y.; Tai, W.L. A high-quality image authentication scheme for AMBTC-compressed images. *KSII Trans. Internet Inf. Syst.* **2014**, 4588–4603. [CrossRef]
11. Zhong, H.; Liu, H.; Chang, C.C.; Lin, C.C. A novel fragile watermark-based image authentication scheme for AMBTC-compressed images. *J. Inf. Hiding Multimedia Signal Process.* **2016**, *7*, 2073–4212.
12. Li, W.; Lin, C.C.; Pan, J.S. Novel image authentication scheme with fine image quality for BTC-based compressed images. *Multimedia Tools Appl.* **2016**, *75*, 4771–4793. [CrossRef]
13. Chen, T.H.; Chang, T.C. On the security of a BTC-based-compression image authentication scheme. *Multimedia Tools Appl.* **2018**, *77*, 12979–12989. [CrossRef]
14. Zhang, X.; Wang, S. Efficient steganographic embedding by exploiting modification direction. *IEEE Commun. Lett.* **2006**, *10*, 781–783. [CrossRef]
15. Hong, W. Adaptive image data hiding in edges using patched reference table and pair-wise embedding technique. *Inf. Sci.* **2013**, *221*, 473–489. [CrossRef]
16. Hong, W.; Chen, T.S.; Chen, J. Reversible data hiding using delaunay triangulation and selective embedment. *Inf. Sci.* **2015**, *308*, 140–154. [CrossRef]
17. Chen, J. A PVD-based data hiding method with histogram preserving using pixel pair matching. *Signal Process. Image Commun.* **2014**, *29*, 375–384. [CrossRef]
18. Chen, H.; Ni, J.; Hong, W.; Chen, T.S. High-fidelity reversible data hiding using directionally-enclosed prediction. *IEEE Signal Process. Lett.* **2017**, *24*, 574–578. [CrossRef]
19. Hong, W.; Chen, T.S. A novel data embedding method using adaptive pixel pair matching. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 176–184. [CrossRef]
20. Malik, A.; Sikka, G.; Verma, H.K. An AMBTC compression based data hiding scheme using pixel value adjusting strategy. *Multidimens. Syst. Signal Process.* **2017**. [CrossRef]
21. Hu, Y.C.; Choo, K.K.; Chen, W.L. Tamper detection and image recovery for BTC-compressed images. *Multimedia Tools Appl.* **2017**, *76*, 15435–15463. [CrossRef]
22. Malik, A.; Sikka, G.; Verma, H.K. A high payload data hiding scheme based on modified AMBTC technique. *Multimedia Tools Appl.* **2017**, *76*, 14151–14167. [CrossRef]
23. Hong, W.; Zhou, X.; Weng, S. Joint adaptive coding and reversible data hiding for AMBTC compressed images. *Symmetry* **2018**, *10*, 254. [CrossRef]
24. Menezes, A.J.; Van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996.
25. The USC-SIPI Image Database. Available online: http://sipi.usc.edu/database/ (accessed on 30 June 2018).