mathematics

MDPI

Article

A High Fidelity Authentication Scheme for AMBTC Compressed Image Using Reference Table Encoding

Tungshou Chen [1], Xiaoyu Zhou [2], Rongchang Chen [3], Wien Hong [1,*] and Kiasheng Chen [4]

[1] Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung City 404, Taiwan; tschen@nutc.edu.tw
[2] School of Information Engineering, Jimei University, Xiamen 361021, China; xiaoyuzhou68@outlook.com
[3] Department of Distribution Management, National Taichung University of Science and Technology, Taichung City 404, Taiwan; rcchens@nutc.edu.tw
[4] School of Electrical and Computer Engineering, Nanfang College of Guangzhou, Guangzhou 510970, China; chenksh@nfu.edu.cn
* Correspondence: wienhong@nutc.edu.tw

Abstract: In this paper, we propose a high-quality image authentication method based on absolute moment block truncation coding (AMBTC) compressed images. The existing AMBTC authentication methods may not be able to detect certain malicious tampering due to the way that the authentication codes are generated. In addition, these methods also suffer from their embedding technique, which limits the improvement of marked image quality. In our method, each block is classified as either a smooth block or a complex one based on its smoothness. To enhance the image quality, we toggle bits in bitmap of smooth block to generate a set of authentication codes. The pixel pair matching (PPM) technique is used to embed the code that causes the least error into the quantization values. To reduce the computation cost, we only use the original and flipped bitmaps to generate authentication codes for complex blocks, and select the one that causes the least error for embedment. The experimental results show that the proposed method not only obtains higher marked image quality but also achieves better detection performance compared with prior works.

Keywords: AMBTC; authentication; detection; PPM

Citation: Chen, T.; Zhou, X.; Chen, R.; Hong, W.; Chen, K. A High Fidelity Authentication Scheme for AMBTC Compressed Image Using Reference Table Encoding. *Mathematics* **2021**, *9*, 2610. https://doi.org/10.3390/math9202610

Academic Editor: Javier Martínez

Received: 4 September 2021
Accepted: 11 October 2021
Published: 16 October 2021

1. Introduction

With the rapid development of the Internet, image editing softwares, such as Photoshop, Snapseed and Mix, have been rapidly developed. These software programs allow people to edit images easily, which means that digital images are vulnerable to be tampered during transmission. Under this trend of Internet development, authenticating the integrity of digital images is gradually becoming an important issue. Fragile watermarking method is a commonly used authentication technique by embedding the authentication code into the image in order to provide protection [1–4]. If the image is tampered during transmission, then the authentication code embedded in the image will also be changed. Therefore, it is possible to know whether an image has been tampered with by determining whether the embedded authentication code has been changed.

Depending on the domain in which the authentication code is embedded, fragile watermarking can be divided into two types, spatial domain and compressed domain. The methods of spatial domain [5–8] embed the authentication code directly into pixels of the image, this type of technique allows to obtain a high image quality and a large space for carrying information. The image authentication methods of the compressed domain [9–22], on the other hand, embed the authentication code into the compressed code. Compressed images are preferred for transmission over the network because smaller files facilitate network transmission. Currently, the joint photographic experts group (JPEG) [9,10], vector quantization (VQ) [11,12], absolute moment block truncation coding (AMBTC) [13–22]

and so on are commonly used compression techniques. Among these techniques, AMBTC requires less computation and has a satisfactory compression rate, which has attracted some research in this area in recent years [13–22].

AMBTC [23], proposed by Lema and Mitchell in 1984, is an improved version of block truncation coding (BTC) [24]. It compresses the image blocks into trios, each trio consisting of two quantization values and a bitmap. The existing AMBTC authentication methods typically embed the authentication codes into the quantized values or bitmaps. Based on the AMBTC compression method, [15] uses a pseudo random number generator (PRNG) to generate the authentication codes and embeds them into the quantized values. Their method is effective in protecting the image to some extent, but the authentication codes can only in base 7. Ref. [16] also proposes a method for AMBTC image authentication based on a special designed reference table, and this method outperforms previous works in terms of marked image quality. However, their approach does not utilize bitmaps to generate authentication codes, resulting in the inability to detect bitmap tampering. Ref. [17] proposes an authentication method with higher security based on the weaknesses of [16]. Ref. [17] generates the authentication code by performing exclusive-or operation of the bitmap with random numbers, which not only maintains the same image quality as in [16], but also detects tampering of the bitmap. To obtain a better authentication effect and marked image quality, [18] uses quantized values of most significant bites (MSB) and bitmaps to generate authentication codes and embeds them into quantized values of least significant bits (LSB). Their method also perturbs the MSB of the quantized values to reduce the error of embedding. Thus, [18] not only preserves the image effectively, but the image quality is also comparable to the method of [17]. To detect the splice tampering, [20] utilizes the block location information in addition to the bitmap to generate the authentication code. They generate different authentication codes by toggling the bitmap and select the authentication code that causes the least error to embed into the quantized values. However, their method does not considered the flipped bitmap may result in a better embedding performance, thus limiting the image quality improvement.

Ref. [19] combines the bitmap with a pseudo-random sequence to generate a 3-bit authentication code, which is embedded into the quantized values by surveying the reference table. Also, this method uses an iterative embedding mechanism to solve the problem that the high quantized values might be smaller than the low ones after embedding. Due to the relatively compact arrangement of the numbers inside reference table, their method achieves a better marked image quality. However, based on the limitations of the searching method, this method can only embed an octal (3 bits) authentication code for each pixel pair. In addition, their searching approach may not find the best marked quantized values to reduce the embedding error. Finally, the authentication code of this method is generated independently of the location information, which makes the swapping of image blocks undetectable.

The existing methods [15–20] can achieve most of the tampering detection and a good image quality. However, due to the design of the embedding technique, the authentication code in [15] can only be digits of base 7, [16,17] can only be digits of bases 2, 4, 8 and 16, while [19] can only be digit of base 8. Furthermore, the authentication code generation of [15,16], [18,19] is independent of the bitmaps or block location information, causing these methods unable to detect some specific malicious tampering. Compared to [15–18], the image quality obtained by [19] is the highest, but the searching approach of this method limits the enhancement of image quality. In contrast to [19], the method of [20] uses a filtering mechanism to reduce the error and obtains a better marked image quality. Nevertheless, the filtering mechanism in [20] does not take into account some characteristics of AMBTC codes, which means that the image quality can be further improved.

For an AMBTC code, if the quantized values are swapped and the bitmap is flipped, the decompressed block will be exactly the same as the one decompressed from the original code [25]. Based on this attractive property, the proposed method toggles one bit of the original and flipped bitmaps in sequence to generate a set of authentication codes.

By surveying the reference table, the authentication code that causes the least error is embedded in quantized values using the PPM technique. Experimental results show that our method not only embeds authentication codes of arbitrary length and obtains the highest image quality, but also achieves a satisfactory detection results compared with the aforementioned methods.

The rest of the paper is structured as follows. Section 2 will introduce the AMBTC compression and the reference table based (RT-based) technique. Section 3 will introduce the proposed method, and experimental results and conclusions will be given in Sections 4 and 5, respectively.

2. Related Works

This section will briefly introduce the basic concepts of AMBTC and the RT-based embedding techniques required by [19] and the proposed method. The details will be presented in the following two subsections.

2.1. AMBTC Compression Method

To encode the image I using the AMBTC compression method, firstly we divide I into N non-overlapping blocks $\{I_i\}_{i=0}^{N-1}$ of size $n \times n$, and calculate the average value m_i of block I_i:

$$m_i = \frac{1}{n \times n} \sum_{j=0}^{n \times n-1} I_{i,j}, \tag{1}$$

where $I_{i,j}$ represents the j-th pixel of i-th block. Next, the value of the j-th bit $B_{i,j}$ in the bitmap B_i is obtained by comparing all the pixels of I_i with m_i:

$$B_{i,j} = \begin{cases} 0 & \text{if } I_{i,j} \leq m_i, \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Based on B_i, the low quantized value a_i and the high quantized value b_i can be calculated by:

$$a_i = \frac{1}{p} \sum_{B_{i,j}=0} I_{i,j}, \tag{3}$$

$$b_i = \frac{1}{q} \sum_{B_{i,j}=1} I_{i,j}, \tag{4}$$

where p and q are the number of '0' and '1' in B_i, respectively. All blocks are processed in the same procedures, and the compression codes $\{a_i, b_i, B_i\}_{i=0}^{N-1}$ of the image I is obtained. The decompressed block D_i can be obtained by decoding $\{a_i, b_i, B_i\}$ using the equation

$$D_{i,j} = \begin{cases} a_i & \text{if } B_{i,j} = 0, \\ b_i & \text{otherwise,} \end{cases} \tag{5}$$

where $D_{i,j}$ is the j-th pixel of D_i. All codes $\{a_i, b_i, B_i\}_{i=0}^{N-1}$ are decompressed in the same manner, and we eventually obtain the decompressed image $D = \{D_i\}_{i=0}^{N-1}$.

Here we briefly introduce the encoding and decoding procedures of AMBTC using a block. Assume a block of size 4×4 is $I_i = [66, 85, 87, 62; 91, 79, 86, 97; 85, 57, 56, 69; 52, 72, 97, 43]$. The mean $m_i = 74$ of I_i is calculated firstly. Next, According to Equations (2)–(4), we obtain $B_i = [0, 1, 1, 0; 1, 1, 1, 1; 1, 0, 0, 0; 0, 0, 1, 0]$, $a_i = 60$, and $b_i = 88$. We can use Equation (5) to decode $\{a_i, b_i, B_i\}$, and the decompressed block $D_i = [60, 88, 88, 60; 88, 88, 88, 88; 88, 60, 60, 60; 60, 60, 88, 60]$ can be calculated.

2.2. The Embedding Techniques Based on Reference Table

The RT-based embedding techniques embed a digit of base 2^{α} into each pixel pair by surveying a reference table. The reference table is a matrix of size 256×256 with numbers in the range from 0 to $2^{\alpha} - 1$. The RT-based techniques to be introduced in this section are the turtle shell embedding (TSE) [26] and the pixel pair matching (PPM) techniques [27]. Figure 1a,b show partial reference tables for TSE and PPM, both of which can be used to embed digits of base $2^3 = 8$.

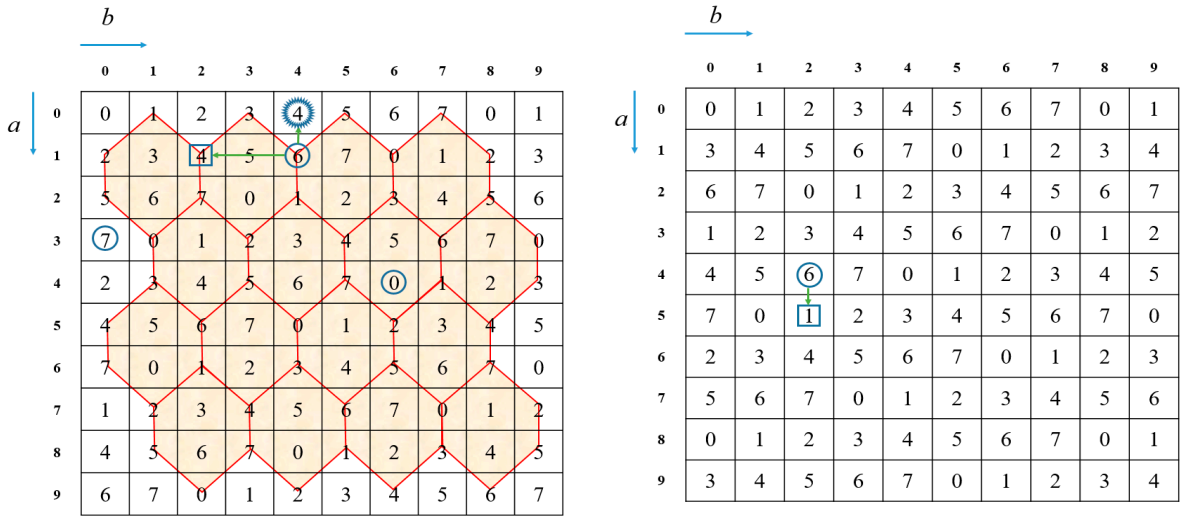


Figure 1. Embedment of TSE and PPM. (**a**) Partial reference table of R_3^{TSE}; (**b**) Partial reference table of R_3^{PPM}.

The TSE embedding method embeds a digit ranging from 0 to $2^3 - 1 = 7$ into a pixel pair (a_i, b_i) by surveying a 3-bit reference table R_3^{TSE}. The value located in (a_i, b_i) in the R_3^{TSE} can be generated by the following equation [28].

$$R_3^{\mathrm{TSE}}(a_i, b_i) = \left[a_i + \left\lceil \frac{1}{2} \times b_i \right\rceil \times 2 + \left\lfloor \frac{1}{2} \times b_i \right\rfloor \times 3 \right] \bmod 8, \tag{6}$$

where a_i and b_i are ranging from 0 to 255. After the reference table R_3^{TSE} is generated, the digits in R_3^{TSE} can be grouped into three categories according to their positions. Different categories use different search methods to embed the authentication codes. In the first case, if (a_i, b_i) is not in the bounds of a complete hexagon (e.g., $(a_i, b_i) = (3, 0)$ in Figure 1a), then we find a location $(\hat{a}_i, \hat{b}_i)$ that is the closest to (a_i, b_i) while satisfying $R_3^{\mathrm{TSE}}(\hat{a}_i, \hat{b}_i) = ac_i$. In the second case, (a_i, b_i) is located in the edge of at least one hexagon, (e.g., $(1, 4)$ or $(2, 2)$ in Figure 1a), then we find a location $(\hat{a}_i, \hat{b}_i)$ from all touched hexagons that is the closest to (a_i, b_i) while satisfying $R_3^{\mathrm{TSE}}(\hat{a}_i, \hat{b}_i) = ac_i$. In the third case, if (a_i, b_i) is within a hexagon (e.g., $(4, 6)$ in Figure 1a), then we find a location $(\hat{a}_i, \hat{b}_i)$ from the hexagon that satisfies $R_3^{\mathrm{TSE}}(\hat{a}_i, \hat{b}_i) = ac_i$. Once the location $(\hat{a}_i, \hat{b}_i)$ is found, ac_i can be embedded by replacing pixel pair (a_i, b_i) with $(\hat{a}_i, \hat{b}_i)$.

However, the searching method provided by the TSE might not always gives the best solution. For example, suppose $ac_i = 4$ is to be embedded into $(a_i, b_i) = (1, 4)$, which is fitted in the second case. Since $R_3^{\mathrm{TSE}}(1, 2) = R_3^{\mathrm{TSE}}(3, 5) = 4$, and $(1, 2)$ is closer to $(1, 4)$, we obtain the embedded pixel pair $(\hat{a}_i, \hat{b}_i) = (1, 2)$. Obviously, $(0, 4)$ is a better solution for this problem because it is closer to $(1, 4)$ than $(\hat{a}_i, \hat{b}_i) = (1, 2)$ and also satisfies $R_3^{\mathrm{TSE}}(0, 4) = 4$. A similar problem also might happen in the third case.

In contrast to the TSE technique, the PPM embedding method also adopts a reference table R_α^{PPM} to embed a digit of base 2^α into each pixel pair, and the base of digit is not limited to 8. The reference table R_α^{PPM} can be generated by using the formula

$$R_\alpha^{\mathrm{PPM}}(a_i, b_i) = (a_i \times c_\alpha + b_i) \bmod 2^\alpha, \tag{7}$$

where c_α is a constant. Some commonly used constants are $c_2 = 2$, $c_3 = 3$, $c_4 = 6$ and $c_6 = 14$. See [27] for a complete list of c_α. To embed ac_i into a pixel pair (a_i, b_i), the PPM method searches in the vicinity of (a_i, b_i) and finds a location $(\hat{a}_i, \hat{b}_i)$ that is closest to (a_i, b_i) while satisfying $R_\alpha^{\mathrm{PPM}}(\hat{a}_i, \hat{b}_i) = ac_i$. During the extraction, reference table R_α^{PPM} and $(\hat{a}_i, \hat{b}_i)$ are known, and the embedded authentication code can be extracted by $ac_i = R_\alpha^{\mathrm{PPM}}(\hat{a}_i, \hat{b}_i)$. Since the PPM method is more flexible and provides a satisfactory embedding performance, the PPM will be used as the embedding technique in our method.

Here is a simple example to illustrate the PPM. Let R_3^{PPM} be a reference table for embedding digits of base $2^3 = 8$. Suppose $ac_i = 1$ and $(a_i, b_i) = (4, 2)$. Since $R_3^{\mathrm{PPM}}(5, 2) = 1$ and $(5, 2)$ is the closest to $(4, 2)$ (see Figure 1b), the marked pixel pair $(\hat{a}_i, \hat{b}_i) = (5, 2)$ is obtained. The embedded authentication code ac_i can be extracted by $R_3^{\mathrm{PPM}}(5, 2) = 1$.

3. The Proposed Method

The authentication codes generated by [15,16,18] are independent of bitmaps or location information; therefore, these methods cannot detect tampering of bitmaps or swapping of image blocks. Ref. [17] can detect some tampering missed by the above methods, but based on the design of the embedding method, the length of the authentication code and the quality of the marked image are limited. Among these methods, only [18] can embed authentication code of arbitrary length. Moreover, the methods of [15–17] are not designed with a selection mechanism to reduce the embedding error. In [18], although the embedding error is reduced by perturbing the MSB of the quantized values, this method uses the LSB replacement which creates a large error, and therefore also limits the quality improvement. To enhance the image quality, [19] employs the TSE to embed the authentication codes. The quality of marked images obtained by [19] is higher than that of [15–18], due to the compact arrangement of digits in the reference table used by TSE. However, according to the analysis in Section 2.2, it can be known that the best marked quantized values may not be found by using the search method of TSE. Moreover, due to the search method, TSE can only work with a 3-bit reference table to embed an authentication code of base 8. As in [18], the generation of authentication codes in [19] is independent of the location information, so the swapping of blocks cannot be detected.

In this paper, we use different approaches to embed authentication codes into smooth and complex blocks. Given a trio $\{a_i, b_i, B_i\}$ of a block D_i, the smoothness of D_i is determined by $|a_i - b_i|$. If $|a_i - b_i| \le T$, where T is a pre-defined threshold, the block is classified as a smooth one; otherwise, it is a complex one. It is known that if the quantized values are swapped and the bitmap is flipped, the decompressed block will be exactly the same as the one decompressed from the original trio [25]. Based on this property, the proposed method toggles a bit in the original and flipped bitmaps of a smooth block to generate a set of authentication codes. These codes are embedded into the quantized values using the PPM, and the trio of the least distorted block after embedding the authentication code is selected as the marked trio. For a complex block, instead of using the toggling technique, only the original and flipped bitmaps are used to generate two authentication codes to reduce the computation cost.

3.1. Embedment of Smooth Blocks

Let $\{a_i, b_i, B_i\}$ be the trio of a smooth block to be embedded with authentication codes, and D_i be the image block decoded from $\{a_i, b_i, B_i\}$. According to the precious property of the AMBTC method, D_i can also be obtained by decoding $\{b_i, a_i, \overline{B}_i\}$, where $\overline{B}_i$ is B_i with all bits flipped. Let $a_i^0 = a_i$, $b_i^0 = b_i$, $B_i^0 = B_i$, $a_i^1 = b_i$, $b_i^1 = a_i$, and $B_i^1 = \overline{B}_i$, then

$\{a_i^0, b_i^0, B_i^0\}$ and $\{a_i^1, b_i^1, B_i^1\}$ can be expressed as $\left\{a_i^f, b_i^f, B_i^f\right\}$, where $0 \le f \le 1$. Let $B_i^{f,k}$ be the bitmap of B_i^f with k-th bit toggled. Using MD5 [29] to hash $\left\{\left\{B_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{n\times n}$ and position information i, and folding each result into α bits, we obtain $2(n \times n + 1)$ candidates of authentication codes $\left\{\left\{ac_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{n\times n}$. The PPM method described in Section 2.2 is then applied to embed these authentication codes into pixel pairs $\left\{a_i^f, b_i^f\right\}_{f=0}^{1}$ and we obtain $\left\{\left\{\hat{a}_i^{f,k}, \hat{b}_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{n\times n}$. $\left\{\left\{\hat{a}_i^{f,k}, \hat{b}_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{n\times n}$ together with $\left\{\left\{B_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{n\times n}$ can be decoded to obtain image blocks $\left\{\left\{\hat{D}_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{n\times n}$. The Euclidian distances between $\left\{\left\{\hat{D}_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{n\times n}$ and D_i are calculated, and the one that has the shortest distance is found and is denoted by $\hat{D}_i^{f^*,k^*}$. The trio of $\hat{D}_i^{f^*,k^*}$, denoted by $\left\{\hat{a}_i^{f^*,k^*}, \hat{b}_i^{f^*,k^*}, B_i^{f^*,k^*}\right\}$, is the marked trio of our method. The procedures of finding the marked trio can be formulated as an optimization problem described below:

$$\text{Minimize}: \sum_{j=0}^{n\times n-1} (\hat{D}_{i,j}^{f,k} - D_{i,j})^2, \tag{8}$$

$$\text{Subject to}: ac_i^{f,k} = \text{hash}_\alpha(B_i^{f,k}, i), \tag{9}$$

$$R_\alpha^{\text{PPM}}(\hat{a}_i^{f,k}, \hat{b}_i^{f,k}) = ac_i^{f,k}, \tag{10}$$

$$\hat{D}_i^{f,k} = \text{de}(\hat{a}_i^{f,k}, \hat{b}_i^{f,k}, B_i^{f,k}), \tag{11}$$

$$0 \le f \le 1,\ 0 \le k \le n \times n,\ k, f \in \text{Integer}, \tag{12}$$

where de() is the decode function of AMBTC, and $\text{hash}_\alpha(x)$ is the function to acquire α-bit authentication code.

Here is an example to illustrate the process of generating authentication codes using MD5. Let α= 4, n = 2, and the bitmap of the eighth block is B_8 = [1, 1, 0, 0]. We use the position i = 8 of the image block and the bitmap B_8 to perform the MD5 operation and obtain 128 bits hash code. Suppose the hex-decimal value of the 128 bits is '36b4c42c4c30d841672290f28e66186a'. Then, we fold the 128 bits in half. The first 64 bits are xor-ed with the remaining 64 bits, and we obtain the 64-bit xor-ed result (the hex-decimal value is '519654dec256c02b'). Repeat this procedure and finally we can obtain a 4-bit authentication code '0001'. In case it is not possible to obtain exactly 4 bits, the extra bits can be discarded. Note that the function $\text{hash}_\alpha(B_i^{f,k}, i)$ used in this paper already contains the folding operation, which can output α-bit authentication code directly.

Next, we use an example to illustrate the embedding procedures of a smooth block D_i. Assume n = 2, T = 6, α = 4 and $\{a_i, b_i, B_i\}$ = {74, 76, 1001}. Since $|a_i - b_i| = 2 < T$, and we have $(a_i^0, b_i^0) = (74, 76)$, $\left\{B_i^{0,k}\right\}_{k=0}^{4}$ ={0001, 1101, 1011, 1000, 1001}, $(a_i^1, b_i^1) = (76, 74)$ and $\left\{B_i^{1,k}\right\}_{k=0}^{4}$ = {1110, 0010, 0100, 0111, 0110}. Suppose the authentication codes generated from $\left\{B_i^{0,k}\right\}_{k=0}^{4}$, and $\left\{B_i^{1,k}\right\}_{k=0}^{4}$ are $\left\{ac_i^{0,k}\right\}_{k=0}^{4}$ = {2, 1, 10, 14, 3} and $\left\{ac_i^{1,k}\right\}_{k=0}^{4}$ = {9, 10, 11, 8, 2}. The codes $\left\{ac_i^{0,k}\right\}_{k=0}^{4}$ and $\left\{ac_i^{1,k}\right\}_{k=0}^{4}$ are then embedded into (74, 76) and (76, 74) using the PPM. According to the reference table R_4^{PPM} shown in Figure 2, we obtain $\left\{\hat{a}_i^{0,k}, \hat{b}_i^{0,k}\right\}_{k=0}^{4}$ = {(73, 76), (73, 75), (74, 78), (75, 76), (73, 77)} and $\left\{\hat{a}_i^{1,k}, \hat{b}_i^{1,k}\right\}_{k=0}^{4}$ = {(77, 75),

(77, 76), (75, 73), (77, 74), (76, 74)}. The trios $\left\{\left\{\hat{a}_i^{f,k}, \hat{b}_i^{f,k}, B_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{4}$ are then decoded and we obtain $\left\{\left\{\hat{D}_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{4}$. The squared distance between $\left\{\left\{\hat{D}_i^{f,k}\right\}_{f=0}^{1}\right\}_{k=0}^{4}$ and D_i is 11, 4, 24, 3, 4, 4, 15, 4, 5 and 0. Since $\hat{D}_i^{f^*,k^*} = D_i^{1,4}$ has the shortest distance to D_i (distance is 0), and the final marked trio $\{\hat{a}_i^{f^*,k^*}, \hat{b}_i^{f^*,k^*}, B_i^{f^*,k^*}\} = \{76, 74, 0110\}$ can be obtained.

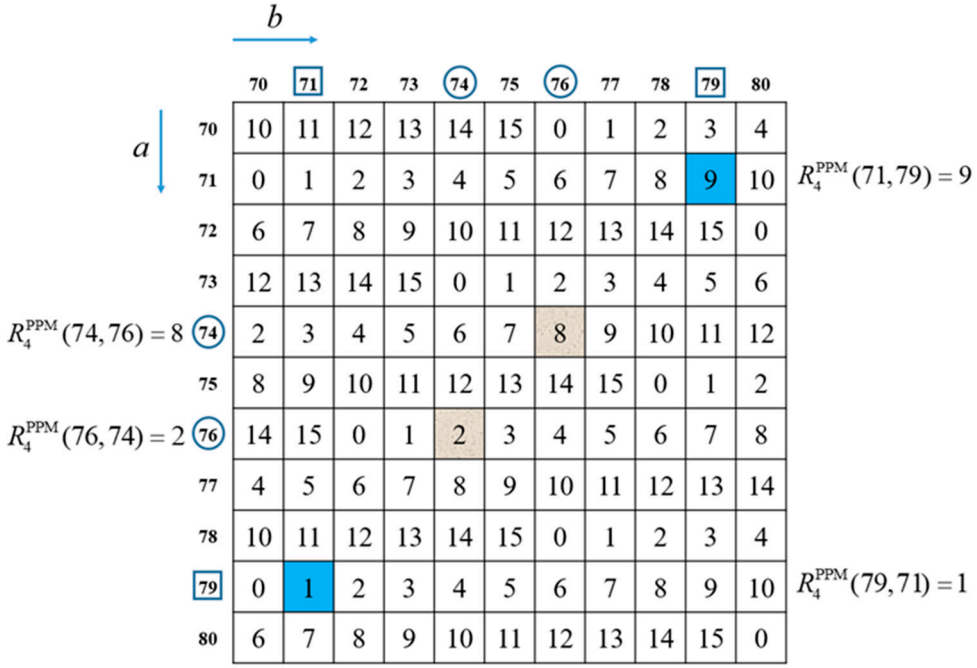


Figure 2. Partial reference table of R_4^{PPM}.

3.2. Embedment of Complex Blocks

It is noted that toggling bits in the bitmaps often causes large errors when the difference between high and low quantized values is large. Thus, for complex blocks, most of the bitmaps of the solutions to Equations (8)–(12) are $B_i^{0,n\times n}$ and $B_i^{1,n\times n}$, i.e., either original bitmap B_i^0 or the flipped bitmap B_i^1 is found in the marked trio. Based on the consideration of computational cost, the toggling technique will not be used in generating authentication codes for complex blocks. Given a trio $\{a_i, b_i, B_i\}$ of a complex block, the location information i and bitmaps $\left\{B_i^f\right\}_{f=0}^{1}$ are used to generate two authentication codes $\left\{ac_i^f\right\}_{f=0}^{1}$ with α bits using Equation (9). Next, the codes $\left\{ac_i^f\right\}_{f=0}^{1}$ are embedded into (a_i^f, b_i^f) using the PPM and we obtain the marked trios $\left\{\hat{a}_i^f, \hat{b}_i^f, B_i^f\right\}_{f=0}^{1}$. Then, decompress $\left\{\hat{a}_i^f, \hat{b}_i^f, B_i^f\right\}_{f=0}^{1}$ to obtain blocks $\left\{\hat{D}_i^f\right\}_{f=0}^{1}$ and find the block $\hat{D}_i^{f^*}$ that has the shorter distance with D_i. The trio of $\hat{D}_i^{f^*}$, denoted by $\left\{\hat{a}_i^{f^*}, \hat{b}_i^{f^*}, B_i^{f^*}\right\}$, is selected as the final marked trio. The embedding procedures of the proposed method can be seen in Figure 3.

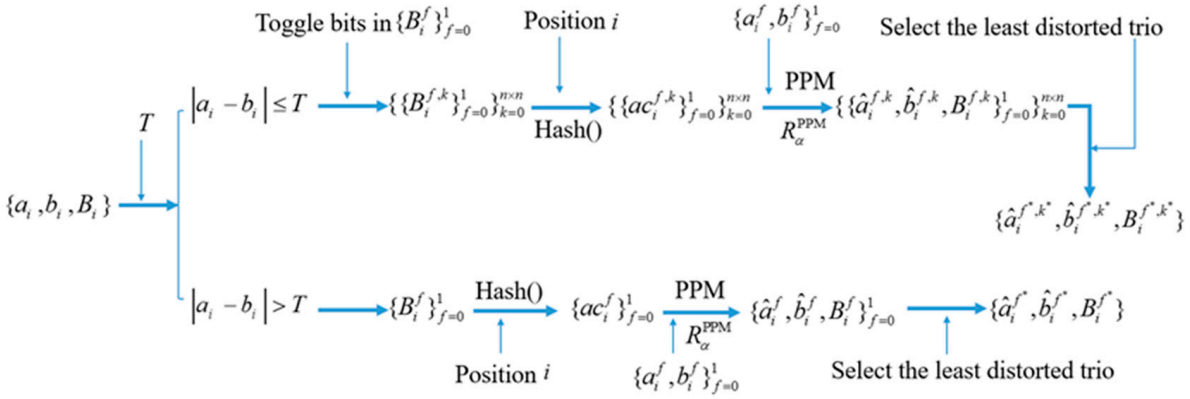


Figure 3. The embedding procedures of the proposed method.

A simple example is presented to show the embedding procedures of a complex block D_i. Let $n = 2$, $T = 6$, $\alpha = 4$ and $\{a_i, b_i, B_i\} = \{71, 79, 1100\}$. Since $|a_i - b_i| = 8 > T$, we have $\{a_i^0, b_i^0, B_i^0\} = \{71, 79, 1100\}$ and $\{a_i^1, b_i^1, B_i^1\} = \{79, 71, 0011\}$. Suppose the authentication codes generated from B_i^0 and B_i^1 are 8 and 12, and embed them into (71, 79) and (79, 71), respectively, using the PPM. According to the reference table R_4^{PPM} shown in Figure 2, we know $R_4^{\mathrm{PPM}}(71, 78) = 8$ and $R_4^{\mathrm{PPM}}(78, 72) = 12$. Therefore, $(\hat{a}_i^0, \hat{b}_i^0) = (71, 78)$ and $(\hat{a}_i^1, \hat{b}_i^1) = (78, 72)$ can be obtained. The trios $\{71, 78, 1100\}$ and $\{78, 72, 0011\}$ are decoded to obtain block $\hat{D}_i^0$ and $\hat{D}_i^1$. Next, the squared distance between $\hat{D}_i^0$ and D_i is 2, while the squared distance between $\hat{D}_i^1$ and D_i is 4. Since $\hat{D}_i^{f^*} = \hat{D}_i^0$ has the shorter distance to D_i, and we obtain the final marked trio $\left\{\hat{a}_i^{f^*}, \hat{b}_i^{f^*}, B_i^{f^*}\right\} = \{71, 78, 1100\}$.

3.3. The Authentication Procedures

This sub-section describes the authentication procedures of the proposed method. Assume that the to-be-authenticated trios are $\{a_i', b_i', B_i'\}_{i=0}^{N-1}$. For each trio $\{a_i', b_i', B_i'\}$, use the function $\mathrm{hash}_\alpha(B_i', i)$ to generate an α-bit authentication code ac_i'. Then, by surveying the reference table R_α^{PPM}, the authentication code eac_i' embedded in (a_i', b_i') is extracted, and determine whether ac_i' and eac_i' are equal. If $ac_i' = eac_i'$, then this block has not been tampered with; otherwise, it has been tampered with. The detailed procedures can be found in Figure 4.

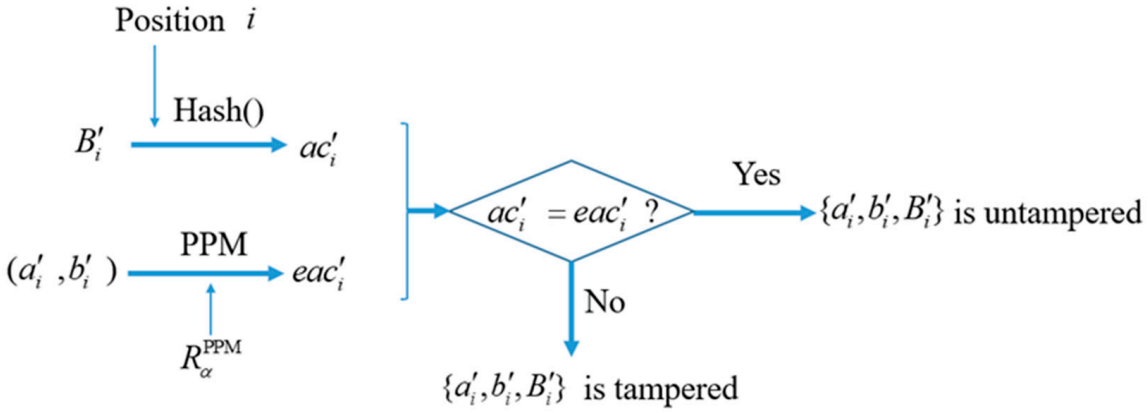


Figure 4. The authentication procedures of the proposed method.

The above authentication procedures are called the first stage authentication. To refine the detection result, the second stage authentication is adopted in the proposed method. Since collisions may occur during authentication, the smaller the length of the embedded

authentication code, the higher the probability of collision. It means that more blocks are tampered with but not detected in the first stage authentication. We know that most of the tampering is clustered in a certain area. Thus, if an untampered block surrounded by blocks that have been tampered with, then the block may has been tampered with as well. Therefore, in the second stage authentication, if a block is detected as untampered in the first stage, but the blocks above and below or left and right of this block have been judged as tampered, the detection result of this block will be modified to be tampered. All blocks are processed using the procedures described above, and the final detection results are obtained.

4. Experimental Results

In order to evaluate the effectiveness of the proposed method, we perform different experiments on a set of grayscale images. Eight grayscale images of size 512×512, including Lena, Tiffany, House, Jet, Peppers, Splash, Boat and Baboon, will be used as test images, as shown in Figure 5. These images can be obtained from the USC-SIPI image database [30]. We also compare the image quality as well as detection results of the proposed method with [17–19]. Image quality is measured using the peak signal-to-noise ratio (PSNR), which is defined as:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2 \times W \times H}{\sum_{i=0}^{W \times H - 1} \left(x_i - x_i' \right)^2} \right), \tag{13}$$

where W and H represent the width and height of a test image, and x_i and x_i' represent the i-th pixel of the AMBTC compressed image and the marked image, respectively. Higher PSNR means better quality of the image. In the following experiments, the block size used is 4×4.

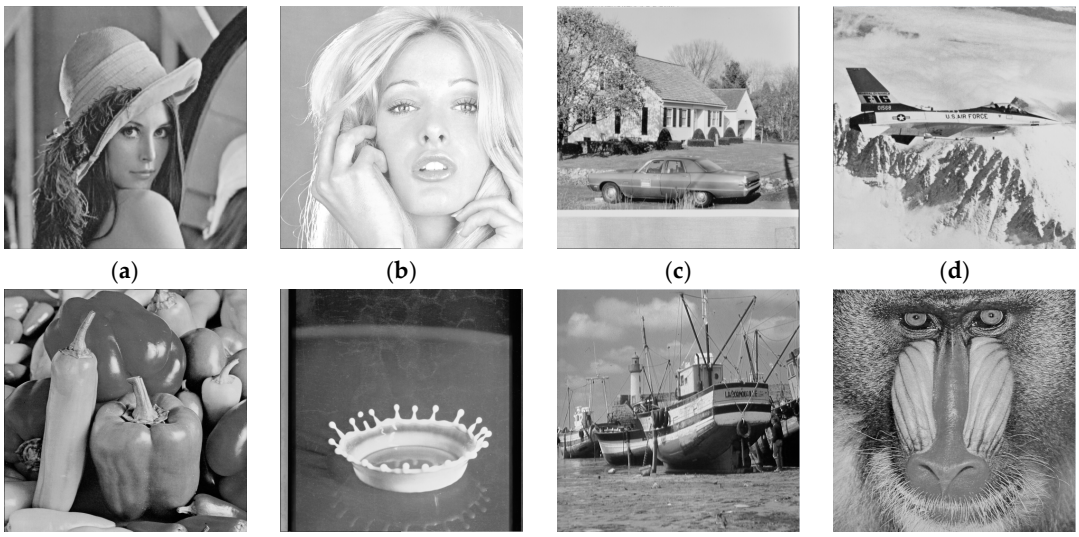


Figure 5. Eight test images. (**a**) Lena; (**b**) Tiffany; (**c**) House; (**d**) Jet; (**e**) Peppers; (**f**) Splash; (**g**) Boat; (**h**) Baboon.

4.1. Quality Evaluation of the Proposed Method

In the proposed method, the computation cost and the marked image quality increase with the increase of threshold T. However, when T exceeds the critical value T^*, it only increases the computational effort and may not contribute to the image quality. Table 1 shows the relationship between threshold value and image quality for embedding authentication code of lengths 2, 4 and 6. It can be found from the table that regardless of α, the image quality is the highest when $T = 255$. The reason is that when $T = 255$, all blocks are

treated as smooth ones. On the contrary, when $T = -1$, all blocks are treated as complex ones, so the obtained quality is the lowest among all thresholds, but less computation is required. Interestingly, when $\alpha = 2$, the quality of the test image is the same at $T = 3$ and $T = 255$, but $T = 3$ requires significantly less computation. Therefore, when $\alpha = 2$, $T^* = 3$ gives the best image quality with affordable computation cost. Similarly, for $\alpha = 4$ and $\alpha = 6$, we choose the thresholds $T^* = 6$ and $T^* = 12$, respectively.

Table 1. Comparisons of image quality for different thresholds when $\alpha = 2, 4, 6$.

α	T	Lena	Tiffany	House	Jet	Peppers	Splash	Boat	Baboon
2	255	54.07	54.20	54.67	54.59	53.99	54.37	54.04	54.05
	3	54.07	54.20	54.67	54.59	53.99	54.37	54.04	54.05
	2	54.06	54.19	54.66	54.58	53.99	54.36	54.04	54.05
	−1	54.00	54.08	54.14	54.07	53.98	54.13	54.01	54.05
4	255	49.35	49. 50	49.73	50.09	49.11	49.75	48.96	48.77
	6	49.35	49. 50	49.73	50.09	49.11	49.75	48.96	48.77
	5	49.34	49.49	49.72	50.08	49.10	49.74	48.95	48.77
	−1	48.76	48.73	48.86	48.83	48.81	48.78	48.77	48.75
6	255	44.81	45.03	44.46	45.41	44.44	45.79	43.77	43.02
	12	44.81	45.03	44.46	45.41	44.44	45.79	43.77	43.02
	11	44.80	45.02	44.46	45.41	44.44	45.78	43.76	43.02
	−1	42.88	42.82	42.91	42.9	42.88	42.84	42.85	42.82

Table 2 lists the suggested threshold T^* in our method with various α. From the table, we can find that as α increases, the threshold value also increases. This is because the more authentication codes a trio embeds, the more damage is done to the quantized values. In this case, a block is more likely needed to toggle a bit in the bitmap to reduce the embedding error. As a result, T^* also becomes larger.

Table 2. Different α corresponds to T^*.

α	2	3	4	5	6	7	8
T^*	3	4	6	8	12	15	20

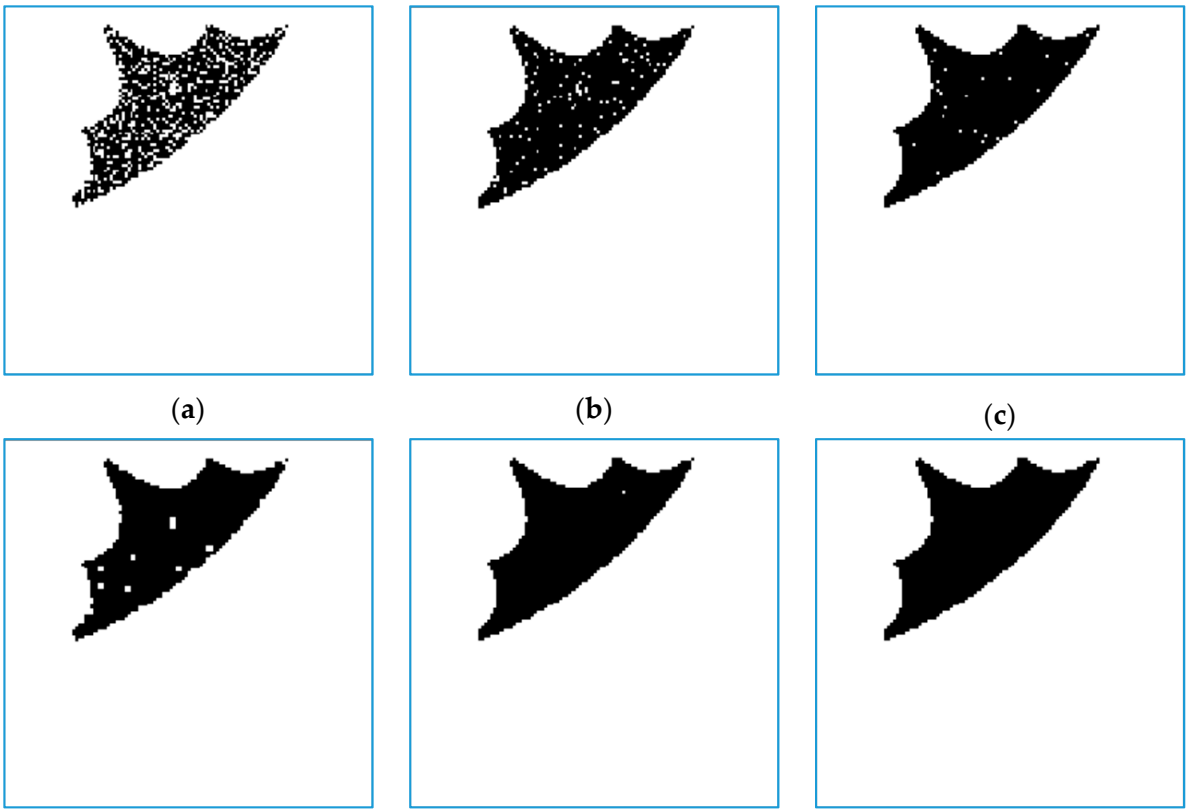
4.2. Detectability of the Proposed Method

This section shows the detectability of the proposed method for the tampered Lena images when α is 2, 4 and 6. First of all, the different lengths of authentication codes are embedded into trios of the Lena image codes to obtain marked images. Figure 2a shows the marked Lena image when $\alpha = 6$. Due to the small number of embedded bits, it is difficult for the human eye to distinguish the marked and original images (See Figure 1a). Next, the marked Lena image is tampered by adding a crown to the Lena's hat, as shown in Figure 6b. There are 16,384 blocks in this image, and 2053 blocks are tampered, which gives a tampering rate of 12.53%.

Figure 7 shows the detection results of the proposed method when $\alpha = 2, 4$ and 6, where Figure 7a–f are the detection results of the first stage and the second stage, respectively. From the first stage detection results, it is clear that the larger the α is, the better the first stage detection will be. This is because the collision probability $1/2^{\alpha}$ when $\alpha = 2, 4$ and 6 is 0.25, 0.0625 and 0.0156, respectively. By comparing Figure 7a–c and Figure 7d–f, it can be found that the second stage detection effectively improves the detection results, especially when α is small.

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

Figure 6. The tampered Lena image. (**a**) Marked image; (**b**) Tampered image; (**c**) Tampered region.



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

(**d**) (**e**) (**f**)

Figure 7. The first and second stage detection results. (**a**) 1st detection, $\alpha = 2$; (**b**) 1st detection, $\alpha = 4$; (**c**) 1st detection, $\alpha = 6$; (**d**) 2nd detection, $\alpha = 2$; (**e**) 2nd detection, $\alpha = 4$; (**f**) 2nd detection, $\alpha = 6$.

To further evaluate the detection performance of the proposed method, different measures will be used to show the detection effectiveness of our method, as shown in Table 3. In the table, true positive rate (TPR) is the probability of a tampered trio also reported as a tampered one, and false negative rate (FNR) is the probability of a tampered trio but reported untampered. As can be seen from the table, the FNRs of the first stage are 25.28%, 5.45% and 1.32% when $\alpha = 2, 4$ and 6, respectively, which are well consistent with the theoretical values of 25%, 6.25% and 1.56%. TPR $= 100\% -$ FNR also agrees with the theoretical value. Interestingly, the second stage detection effectively increased the TPR while decreasing the FNR. For example, when $\alpha = 2$, the probability of TPR in the first stage is 74.72%, which rises to 95.62% in the second stage. Meanwhile, the probability

of FNR dropped from 25.28% to 4.38%. Therefore, it is desirable to use the second stage of detection.

Table 3. TPR and FNR values for the first and second stage detections.

α	2		4		6	
Detection stage	1st	2nd	1st	2nd	1st	2nd
TPR	74.72%	95.62%	94.55%	99.61%	98.68%	99.90%
FNR	25.28%	4.38%	5.45%	0.39%	1.32%	0.10%

4.3. Quality Comparisons with Other Works

To show the superiority of our method, we compare the image quality of the proposed method with [17–19] for $\alpha = 2, 3, 4, 6$, as shown in Table 4. In this table, 'n/a' indicates that the method is not applicable. From the table, we can observe that the method of [17] can embed authentication code of lengths 2, 3 and 4, whereas it cannot embed 6 bits. This is because their method can only embed up to 4 bits. By improving this shortcoming of [17], the method of [18] can embed authentication code of arbitrary length. The image quality obtained by [17] is the lowest among the compared methods when $\alpha = 2$ and 3. In contrast, the image quality of [18] is the lowest when $\alpha = 4$, which is due to the design of their embedding method.

Table 4. Image quality comparisons with other works for $\alpha = 2, 3, 4, 6$.

α	Methods	Lena	Tiffany	House	Jet	Peppers	Splash	Boat	Baboon
2	[17]	49.90	49.90	49.86	49.93	49.89	49.90	49.89	49.86
	[18]	51.75	51.79	51.77	51.77	51.71	51.80	51.76	51.77
	[19]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Proposed	54.07	54.20	54.67	54.59	53.99	54.37	54.04	54.05
3	[17]	46.40	46.49	46.49	46.53	46.43	46.47	46.48	46.49
	[18]	48.56	48.58	48.66	48.64	48.62	48.61	48.62	48.54
	[19]	49.79	49.81	49.99	49.93	49.77	49.83	49.79	49.72
	Proposed	51.68	51.82	52.21	52.28	51.62	51.89	51.58	51.60
4	[17]	46.42	46.38	46.35	46.34	46.36	46.37	46.36	46.40
	[18]	45.79	45.79	45.85	45.77	45.77	45.82	45.74	45.74
	[19]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Proposed	49.35	49. 50	49.73	50.09	49.11	49.75	48.96	48.77
6	[17]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	[18]	39.71	39.75	39.7	39.69	39.7	39.66	39.74	39.73
	[19]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Proposed	44.81	45.03	44.46	45.41	44.44	45.79	43.77	43.02

As for [19], their method can only embed a 3-bit authentication code into each trio based on a reference table R_3^{TSE}. Nevertheless, due to the compact arrangement of digits in R_3^{TSE}, the image quality obtained by [19] is better than that of [17,18]. For example, when $\alpha = 3$, the Lena image quality of [19] is 49.79 dB, while the quality of [17,18] are 46.40 and 48.56 dB, respectively. However, the Lena image quality of the proposed method is 51.68 at $\alpha = 3$, which is higher than that of [19]. In addition, the proposed method can achieve the best image quality regardless of α. For example, when $\alpha = 2$, the Lena image quality of our method is 54.07 dB, which is $54.07 - 49.90 = 4.17$ dB higher than [17] and $54.07 - 51.75 = 2.32$ dB higher than [18]. From the above analysis, it is clear that our

method can not only embed arbitrary length of authentication code, but also obtain the best image quality.

In addition to comparing the PSNR of images, we also compared the structural similarity (SSIM) [31] for different methods using $\alpha = 3$. The results are shown in Table 5. SSIM is a metric for measuring the similarity of images. A larger SSIM means that the two images are more similar. The maximum value of SSIM is 1. As can be seen from the table, the SSIMs of different methods are all greater than 0.980, meaning that a satisfactory visual effect can be obtained by these methods. However, among the compared methods, the proposed method obtains the highest SSIM. Taking the test image Lena as an example, the SSIMs obtained in [17–19] are 0.990, 0.994 and 0.995, respectively, while the SSIM obtained by our method is 0.997. Note that thought the experiments are conducted using $\alpha = 3$, other α also reveals similar results. Therefore, compared with other methods, our method can obtain better visual effects.

Table 5. SSIM comparisons with other works for $\alpha = 3$.

Methods	Lena	Tiffany	House	Jet	Peppers	Splash	Boat	Baboon
[17]	0.990	0.989	0.991	0.989	0.990	0.987	0.993	0.997
[18]	0.994	0.993	0.995	0.993	0.994	0.992	0.995	0.998
[19]	0.995	0.995	0.996	0.995	0.995	0.994	0.997	0.998
Proposed	0.997	0.997	0.998	0.997	0.997	0.996	0.998	0.999

It is worth noting that the proposed method uses MD5 to generate the authentication codes, which will take longer time for code generation compared to the approach used in [17]. However, the MD5 (and other hash function) is sensitive to the input. That is, small alterations in pixel values will generate different codes. The proposed method subtly takes advantage of this feature to generate a set of codes, and the one that causes the least distortion is selected as the final authentication code. Therefore, the image quality of the proposed method outperforms other works.

4.4. Detectability Comparisons with Other Works

This section shows the detectability of [17–19] and the proposed method for various tamperings. Figure 8 shows the tampering of the marked Jet image with three lifting bodies clipped from another image. The marked and tampered images are shown in Figure 8a,b, whereas the corresponding tampered regions are given in Figure 8c. Since the method of [19] can only embed authentication code of length 3, we set $\alpha = 3$ in the experiments of this section for a fair comparison.

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

Figure 8. The tampering of the Jet image. (**a**) Marked image; (**b**) Tampered image; (**c**) Tampered regions.

Figure 9 shows the first and second stage detection results of [17–19] and the proposed method. In these figures, the white dots in the tampered regions represent undetected tampered blocks. Since $\alpha = 3$, the probability of collision rate is approximately $1/2^3 = 0.125$, meaning that around 12.50% tampered blocks are undetected, as shown in Figure 9a–d. However, the white dots are significantly reduced (see Figure 9e–h) if the second stage detection is applied, indicating the effectiveness of using this stage.

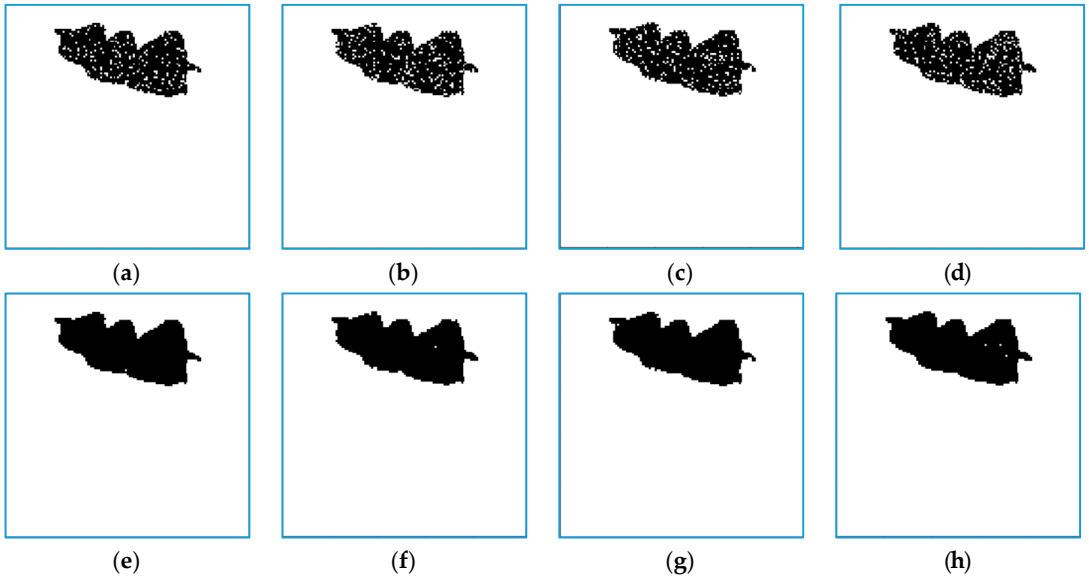


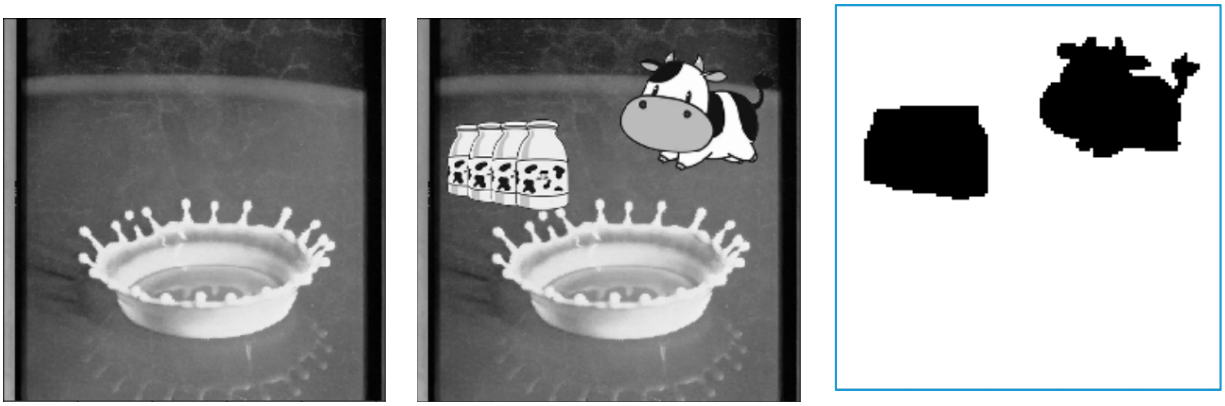
Figure 9. The 1st and 2nd detection results of different methods. (**a**) 1st stage, [17]; (**b**) 1st stage, [18]; (**c**) 1st stage, [19]; (**d**) 1st stage, proposed method; (**e**) 2nd stage, [17]; (**f**) 2nd stage, [18]; (**g**) 2nd stage, [19]; (**h**) 2nd stage, proposed method.

To further evaluate the detection performance of various methods, we use different metrics to measure their effectiveness, as shown in Table 6. In this table, TPR = TP/(TP+FN), FNR = FN/(TP+FN) and PR = TP/(TP+FP). TP indicates the number of blocks that are tampered with and detected as tampered. FN is the number of blocks that are tampered but detected as not tampered, whereas FP represents the number of blocks that are not tampered but detected as tampered. Since $\alpha = 3$, the collision rate should be $1/2^3 = 0.125$, which is consistent with the results shown in Table 6. Since TPR $= 1 -$ FNR, the theoretical TPR is 87.5%. In the first stage detection, the TPR of [17–19] and the proposed method are 89.83%, 87.66%, 88.60% and 87.83%, respectively, which are in accordance with the theoretical value. In the secondary detection, the TPR of all methods are up to 99.00%. In the first detection, the PRs are all 100%, which is because these methods do not misjudge untampered blocks as being tampered. However, in the secondary detection, some untampered blocks are mistakenly judged to have been tampered with. Therefore, the PR is slightly reduced.

To further compare the detectability of the proposed method with related works, we conduct additional experiments using two types of tampering. Firstly, these methods are used to embed authentication codes into the Splash image. Then we tamper the marked images by splicing bottles and a cow. The marked image, tampered image, and tampered regions are shown in Figure 10a–c, respectively. In Figure 10b, we obtain the AMBTC trios of bottles firstly, and then splice the decompressed trios onto the marked image. This type of tampering is referred to as Type A. The tampering of cow is done by replacing blocks with other ones of the marked image that have the shortest Euclidian distance to the corresponding blocks of the cow image. This type of tampering is referred to as Type B. Since the method of [19] can only embed authentication code of length 3, we set $\alpha = 3$ in this experiment for a fair comparison.

Table 6. Comparisons of different methods of TPR, FNR, and PR.

Methods	Stages	TPR	FNR	PR
[17]	1st	89.83%	10.17%	100.00%
	2nd	99.33%	0.67%	99.94%
[18]	1st	87.66%	12.34%	100.00%
	2nd	98.78%	1.22%	99.83%
[19]	1st	88.60%	11.39%	100.00%
	2nd	99.17%	0.83%	99.83%
Proposed	1st	87.83%	12.17%	100.00%
	2nd	99.67%	0.33%	99.78%



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

Figure 10. The tampering of Splash image. (**a**) Marked image; (**b**) Tampered image; (**c**) Tampered regions.

Figure 11 shows the results of the second-stage detection with different methods. For the tampering of bottles, [17–19] and the proposed method can all obtain a better detection result with 99.99% detection rate. However, the methods of [18,19] do not detect the tampering of cow image, as shown in Figure 9b,c. This is because the generation of authentication codes for their methods is independent of the location information. Yet, both [17] and the proposed method are able to detect both types of tampering with better results, as shown in Figure 9a,d.

In addition to comparing with [17–19], we also compare with the existing methods [15,16], as shown in Table 7. In this table, 'Type A' and 'Type B' represent the tampering approaches of bottles and cow in Figure 10, respectively. As can be seen from the table, all compared methods can detect the tampering of bottles, but only [17] and the proposed method can detect the tampering of cow. This is because these two methods take both bitmaps and location information into account when generating authentication codes. Besides, for each trio, we can find that [15] can only embed 2.8 bits (a digit of base 7), [16,17] can only embed at most 4 bits, and [19] can only embed 3 bits (a digit of base 8). As for [18] and our method, it is possible to embed an authentication code of arbitrary length. In addition, the proposed method uses a better embedding and selection mechanisms than other methods, and thus the image quality is also better, as shown in the above experiments.

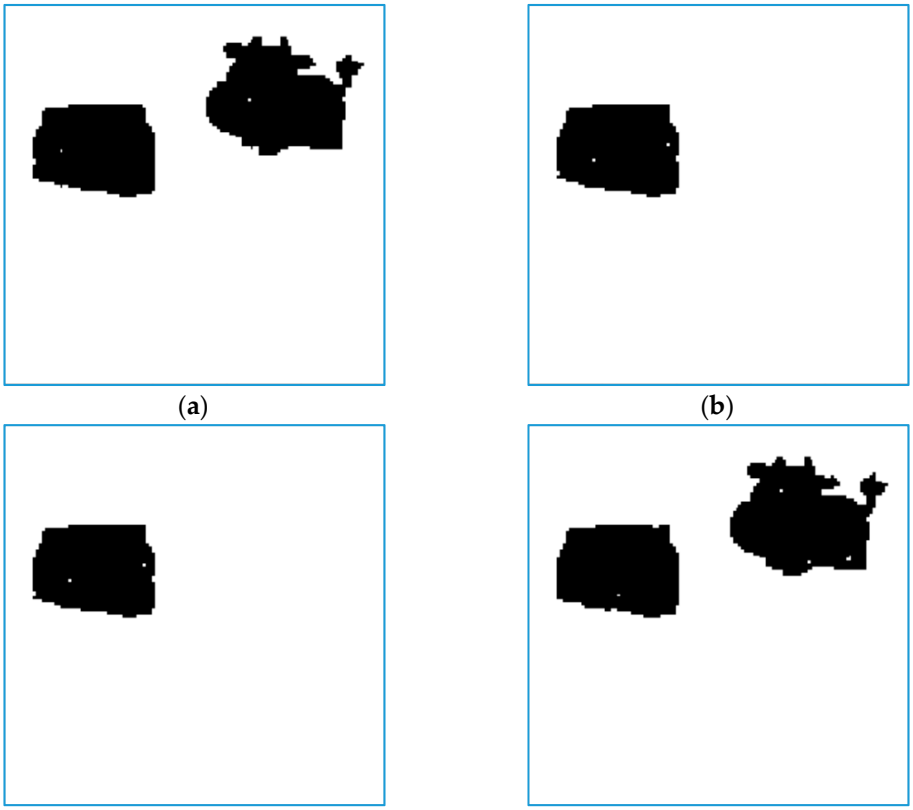


Figure 11. The second stage detection results of different methods. (**a**) Detection result of [17]; (**b**) Detection result of [18]; (**c**) Detection result of [19]; (**d**) Detection result of the proposed method.

Table 7. Comparisons with other works.

Methods	**[15]**	**[16]**	**[17]**	**[18]**	**[19]**	**Proposed**
Generation of AC	Random numbers	Random numbers	Random numbers and bitmap	Quantized values and bitmap	Bitmap	Bitmap and location information
Generator of AC	Random number generator	Random number generator	exclusive-or	Hash function	Hash function	Hash function
Payload of each trio	2.8	1, 2, 3, 4	1, 2, 3, 4	A digit of base 3–256	3	A digit of base 3-256
Detectability of Type A	Yes	Yes	Yes	Yes	Yes	Yes
Detectability of Type B	No	No	Yes	No	No	Yes

5. Conclusions

In this paper, we propose a high quality image authentication method with AMBTC-based authentication. Based on the characteristics of AMBTC, The proposed method generate the authentication codes by using both the original and flipped bitmaps. To improve the image quality, we toggle the bits in the original and flipped bitmaps to generate a series of authentication codes for smooth blocks, and then select the authentication code that causes the least error for embedment. To lower the computational cost, we use a reduced selection mechanism for complex blocks. Moreover, our method uses PPM to embed authentication codes to further enhance the image quality. Experimental results show that our method not only achieves a better tampering detection, but also yields a high marked image quality. The focus of this paper is mainly on improving the detectability and

marked image quality. Future work could include a recoverability feature so that tampered areas can be recovered.

Author Contributions: T.C., X.Z. and W.H. contributed to the conceptualization, methodology, and writing of this paper. R.C. and K.C. conceived the simulation setup, formal analysis and conducted the investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

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

References

1. Bravo-Solorio, S.; Calderon, F.; Li, C.T.; Nandi, A.K. Fast fragile watermark embedding and iterative mechanism with high self-restoration performance. *Digit. Signal Process.* **2017**, *73*, 83–92. [CrossRef]
2. Garg, P.; Jain, A.K. An invisible based watermarking technique for biometric image authentication. *Mater. Today Proc.* **2021**, *46*. [CrossRef]
3. Mohammed, B.; Ameen, S.Y.; Mohammed, O. Image authentication based on watermarking approach: Review. *Asian J. Comput. Sci. Inf. Technol.* **2021**, *9*, 34–51.
4. Qin, C.; Ji, P.; Wang, J.; Chang, C.C. Fragile image watermarking scheme based on VQ index sharing and self-embedding. *Multimed. Tools Appl.* **2017**, *76*, 2267–2287. [CrossRef]
5. Chen, H.; Ni, J.; Hong, W.; Chen, T.S. Reversible data hiding with contrast enhancement using adaptive histogram shifting and pixel value ordering. *Process. Image Commun.* **2016**, *46*, 1–16. [CrossRef]
6. Zenati, A.; Ouarda, W.; Alimi, A.M. A new digital steganography system based on hiding online signature within document image data in YUV color space. *Multimed. Tools Appl.* **2021**, *6*, 18653–18676. [CrossRef]
7. Weng, S.; Pan, J.S. Reversible data hiding based on an adaptive pixel-embedding strategy and two-layer embedding. *Inf. Sci.* **2016**, *369*, 144–159. [CrossRef]
8. Zhou, J.; Sun, W.; Dong, L.; Liu, X.; Au, O.C.; Tang, Y.Y. Secure reversible image data hiding over encrypted domain via key modulation. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 441–452. [CrossRef]
9. Patra, J.C.; Phua, J.E.; Bornand, C. A novel DCT domain CRT-based watermarking scheme for image authentication surviving JPEG compression. *Digit. Signal Process.* **2010**, *20*, 1597–1611. [CrossRef]
10. Zairi, M.; Boujiha, T.; Ouelli, A. Improved JPEG image watermarking in data compression domain using block selection strategy. *EAI Endorsed Trans. Internet Things* **2021**, *6*, e4. [CrossRef]
11. Xu, S.; Horng, J.H.; Chang, C.C. Reversible data hiding scheme based on VQ prediction and adaptive parametric binary tree labeling for encrypted images. *IEEE Access* **2021**, *9*, 55191–55204. [CrossRef]
12. Hong, W.; Zhou, X.Y.; Lou, D.C.; Chen, T.S.; Li, Y. Joint image coding and lossless data hiding in VQ indices using adaptive coding techniques. *Inf. Sci.* **2018**, *463–464*, 245–260. [CrossRef]
13. Uma, P.; Vimala, S. Reversible data hiding technique using AMBTC based bitmap manipulation. *Int. J. Comput. Appl.* **2020**, *176*, 48–55. [CrossRef]
14. Kim, C.; Yang, C.N. Self-embedding fragile watermarking scheme to detect image tampering using AMBTC and OPAP approaches. *Appl. Sci.* **2021**, *11*, 1146. [CrossRef]
15. 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 Multimed. Signal Process.* **2016**, *7*, 2073–4212.
16. Li, W.; Lin, C.C.; Pan, J.S. Novel image authentication scheme with fine image quality for BTC-based compressed images. *Multimed. Tools Appl.* **2016**, *75*, 4771–4793. [CrossRef]
17. Chen, T.H.; Chang, T.C. On the security of a BTC-based-compression image authentication scheme. *Multimed. Tools Appl.* **2018**, *77*, 12979–12989. [CrossRef]
18. Hong, W.; Zhou, X.Y.; Lou, D.C.; Huang, X.Q.; Peng, C. Detectability improved tamper detection scheme for absolute moment block truncation coding compressed images. *Symmetry* **2018**, *10*, 318. [CrossRef]
19. Chen, C.C.; Chang, C.C.; Lin, C.C.; Su, G.D. TSIA: A novel image authentication scheme for AMBTC-based compressed images using turtle shell based reference matrix. *IEEE Access* **2019**, *7*, 149515–149526. [CrossRef]
20. Hong, W.; Li, D.; Lou, D.C.; Zhou, X.Y.; Chang, C.H. A bit toggling approach for AMBTC tamper detection scheme with high image fidelity. *PLoS ONE* **2020**, *15*, e0230997. [CrossRef]
21. Kumar, R.; Kumar, N.; Jung, K.H. Color image steganography scheme using gray invariant in AMBTC compression domain. *Multidimens. Syst. Signal Process.* **2020**, *31*, 1145–1162. [CrossRef]

22. Chen, Y.H.; Chang, C.C.; Lin, C.C.; Wang, Z.M. An adaptive reversible data hiding scheme using AMBTC and quantization level difference. *Appl. Sci.* **2021**, *11*, 635. [CrossRef]
23. Lema, M.; Mitchell, O. Absolute moment block truncation coding and its application to color image. *IEEE Trans. Inf. Forensics Secur.* **1984**, *10*, 507–518. [CrossRef]
24. Delp, E.; Mitchell, O. Image compression using block truncation coding. *IEEE Trans. Commun.* **1979**, *27*, 1335–1342. [CrossRef]
25. Hong, W.; Chen, T.S.; Shiu, C.W. Lossless steganography for AMBTC-compressed images. In Proceedings of the 2008 Congress on Image and Signal Processing, Sanya, China, 27–30 May 2008.
26. Chang, C.C.; Liu, Y.; Nguyen, T.S. A novel turtle shell based scheme for data hiding. In Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding & Multimedia Signal Processing, Kitakyushu, Japan, 27–29 August 2014.
27. 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]
28. Liu, L.; Chang, C.C.; Wang, A. Data hiding based on extended turtle shell matrix construction method. *Multimed. Tools Appl.* **2017**, *76*, 12233–12250. [CrossRef]
29. Menezes, A.J.; Van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996.
30. The USC-SIPI Image Database. Available online: http://sipi.usc.edu/database/ (accessed on 1 July 2021).
31. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]