Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **2**, March **2016**

A Novel Fragile Watermark-Based Image Authentication Scheme for AMBTC-Compressed Images

Hong Zhong

School of Computer Science and Technology
Anhui University
Hefei, China
zhongh@ahu.edu.cn

Haiquan Liu

School of Computer Science and Technology
Anhui University
Hefei, China
haiquanliu01@sina.com

Chin-Chen Chang

Department of Information Engineering and Computer Science
Feng Chia University
Taichung, Taiwan
alan3c@gmail.com

Chia-Chen Lin*

Department of Computer Science and Information Management
Providence University
Taichung, Taiwan
*Corresponding Author: mhlin3@pu.edu.tw

Received May, 2015; revised August, 2015

Abstract. *A novel fragile watermark-based image authentication scheme for images that are compressed by absolute moment block truncation coding (AMBTC) was proposed in this paper. In the proposed scheme, an authentication code is generated by a pseudo random sequence. Subsequently, the authentication code is embedded into both quantization levels and the bit map of each image block for AMBTC-compressed images. Unlike other schemes, it is not necessary to consider various cases and obtain different image quality according to the number of bits of the authentication code in binary form. Experimental results demonstrate that our new scheme provides good image quality for watermarked images. Moreover, the high image quality is not influenced by an increasing authentication code.*
Keywords: Fragile watermark, Image authentication, Absolute moment block truncation coding (AMBTC).

1. **Introduction.** Contemporary society is maturing in terms of informationization and digitization. Users now frequently transmit digital images through the Internet. However, security and efficiency continues to be a concern during the transmission of digital images. Image compression is the most common way of storing and transmitting images

efficiently and is divided into two types: lossless compression [1] and lossy compression [1]. Lossless compression can recover the original images without causing any distortion, and examples include run-length encoding (RLE), Huffman coding, etc. In comparison, lossy compression cannot recover original images perfectly, but it can better compress digital images. Therefore, in most situations, digital images are stored and transmitted in a compressed format.

Currently, there is a diverse assortment of lossy compression techniques such as JPEG [2], block truncation coding (BTC) [1,3,4,25], vector quantization (VQ) [1,6], etc. A typical approach is the conventional BTC designed by Delp and Ritcell [3], which is one of the most widely-used methods due to its low computational cost. Later, Lema and Mitchell [4] improved the BTC scheme through an improved method for calculating two quantization levels for each block. This is called absolute moment block truncation coding (AMBTC) and obtains higher image quality than the conventional BTC.

While image-processing tools have become increasingly powerful, transmission has been largely left untrusted. In other words, the digital images can be easily manipulated, modified or forged by malicious users. Traditional cryptographic schemes [14] are the most direct approach to protect the integrity of images, such as MD5 and RSA [24]. If the image is modified, it will be detected when receivers decrypt the image. However, it is often not possible to locate the tampered areas through these approaches. In other words, traditional cryptography is not suitable for image authentication.In addition, the image data hiding become more mature that can be divided into irreversible data hiding and reversible data hiding [26,27]. For these reasons, image authentication [5] has attracted increasing attention as a promising technology to achieve secure transmission of digital images, the images used to be authenticated can be grayscale images and color images[28].

Next, we discuss two typical schemes of image authentication which are digital-signature based schemes [19,20] and fragile watermarking based schemes [15-18]. In a digital-signature based scheme, There are three important processes for a signature-based method. First of all, a hash digest is generated by a hash function which is used to process the given image. Then, the hash digest is signed by the digital signature algorithm [24] . Then the digital signature of the image is stored by a trusted third party or attached to the image. The last step, when the image is to be authenticated, the new digital signature is generated based on the current image and it is compared to the original signature, which maybe extracted from the image or the trusted third party. Generally speaking, it is necessray to store and transmit additional imformation by this kind of scheme although the tampered areas can be detected.

The fragile watermark-based scheme can avoid the problem of the additional costs of space [15-18]. In this scheme, it is also divided into three main steps: watermark data generation, watermark data embedding and watermark data extraction. In general, the watermark data are generated according to the features of images, the contents of images or the random values which both the sender and the receiver know in advance. After that, we embed the watermark data into the original images in many different ways. Finally, receivers extract the watermark data from the given image and examine image integrity when they authenticate the image. If the generating of watermark data relates to the original image, we need to make sure that there is no difference in the original content or features after the embedding manipulation.

In 2001, Lin and Chang [22] presented a semi-fragile watermarking technique that can prevent malicious manipulations, but allow for lossy compression of JPEG images. Wong and Memon described a secret and public key watermarking method for ownership verification and authentication of digital images [21]. In 2006, a high embedding efficiency steganographic method was proposed by Zhang et al. [8] and we used the novel function

in our new scheme. In 2008, an effective dual watermark scheme for tamper detection and recovery of digital images was proposed by Lee and Lin [9]. In 2010, Ahmed and Siyal designed a hash-based authentication scheme [10] that can address several issues simultaneously such as tamper detection, security and robustness. In 2011, an adaptive image authentication scheme [11] was introduced by Chung and Hu that is suitable for VQ-compressed images. A quantization-based semi-fragile watermark [12] was introduced by Qi and Xin that is suitable for image content authentication. Zhang et al. [23] proposed a novel watermarking scheme that can achieve flexible self-recovery. Next, a tamper detection and self-recovery scheme for biometric images was proposed by Li et al. in 2012 [13].

In 2013, Hu et al. [15] proposed a scheme of image authentication that is used for AMBTC-compressed images; however, the image quality of the watermarked images was less than ideal. Lin et al. [16] improved upon this through a high-quality image authentication scheme in 2014. However, when the watermarking data is more than two bits in a binary format and the corresponding decimal number is three, the image quality for the watermarked image are unacceptable. As a result, we propose a novel image authentication scheme for images that are compressed by AMBTC to achieve a higher image quality and maintain the detection capability for a watermarked image even when the authentication code is more than two bits in a binary format. In other words, the authentication code is greater than three in decimal format. In the proposed scheme, the authentication code is embedded into two quantization levels and the bit map of each AMBTC-compressed image block. The experimental results demonstrated that our scheme does better than Lin et al.s scheme in terms of image quality.

This paper is structured as follows: we review related work in Section 2. The proposed scheme is introduced in Section 3. Section 4 presents the experiments and discussions. Finally, we conclude our scheme in Section 5.

2. **Related Work.**

2.1. **Absolute moment block truncation coding (AMBTC).**

A block truncation coding scheme was first proposed by Delp and Ritcell [3] in 1979. In order to decrease the mean squared error, Lema and Mitchell [4] proposed the concept of absolute moment block truncation coding (AMBTC) whose mean squared error has been proven to be the minimal above all BTC schemes. The schemes are almost similar, apart from the equation for computing the two quantization levels.

Both conventional BTC and the AMBTC deal with images by block. In other words, the original image is divided into many non-overlapping blocks whose size are $n \times n$ and it is denoted as k. Then, these original image blocks are orderly processed. Firstly, it is necessary to calculate the mean value $\bar{x}$ of each image block that contains $n \times n$ pixels. Then, it classifies these pixels into two groups according to their value. If the value of one pixel is greater than $\bar{x}$, it is classified into the second group which is denoted by G_1. Otherwise, it is classified into the first group, which is denoted by G_0. A corresponding bit in the bit map will be stored as value 0 or 1 if this pixel is classified into G_0 or G_1, respectively.

The next step is to calculate the two quantization levels l and h for each block and they are calculated according to the following two equations:

$$l = \frac{1}{k-p} \times \sum_{x_i < \bar{x}} x_i,$$

$$h = \frac{1}{p} \times \sum_{x_i \geq \bar{x}} x_i.$$

Here x_i stands for the i th pixel of the block in the order of left-to-right and top-to-down, and p denotes the number of pixels in the second group G_1. Finally, a block will be compressed as a trio (l, h, BM).

In AMBTC, the image decoding procedure takes the compressed trio (l, h, BM) as input and outputs the reconstructed image block for each block of the image. If the bit value is 0 in the bit map, the corresponding pixel is recovered by the quantization level l. Otherwise, it is reconstructed by the quantization level h.

2.2. **Lin et al.s scheme.** Recently, Lin et al. proposed an authentication scheme for images that have been compressed by AMBTC in advance [16]. It is a fragile watermark-based scheme that makes extensive use of the trio (l, h, BM) for an image block. Specifically, they used the bit map of the block to generate an authentication code (acu). After that, the acu is embedded into the two quantization levels l and h of this image block. Finally, it is necessary to extract the $eauc$ from the watermarked images when we want to authenticate it.

Simply speaking, we just consider the block size as 4×4. The trio of a block is converted into binary format and they are denoted as $l = (l_0, l_1, l_2, l_3, l_4, l_5, l_6, l_7)$, $h = (h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7)$ and $BM = (BM_0, BM_1, ..., BM_{15})$, respectively. Before the generation of auc, the elements of BM are divided into some groups according to the range of auc. Here, the number of groups equals to the number of bits in auc which is defined as eb. Then, each group contributes a bit to generate the complete authentication code. For security reasons, there are two possible positions (pc) to embed authentication data. To reduce the difference after embedding, when the last bit of authentication code auc_1 is 0, they embed auc_1 into the low level. Otherwise, it is embedded into the high level. As a result, there are four possible ways to embed an authentication code. As an example to explain the process of auc embedding in detail, let us suppose that $eb = 2$. The corresponding equation for embedding is as follows:

$$l_{pc} = auc_1, h_7 = auc_2, \quad if\, auc_1 = 0,$$
$$h_{pc} = auc_1, l_7 = auc_2, \quad otherwise.$$

The four possible ways are $(h_7, l_6), (l_7, h_6), (h_7, l_7), (l_7, h_7)$ to embed the two bits of $auc = (auc_2, auc_1)$. As mentioned above, there are two kinds of possible positions which are 6 and 7. Note that the first two situations correspond to $pc = 6$ and the remaining two correspond $pc = 7$. If the generated authentication code is 10 and $pc = 6$, it will choose the first situation and the process of embedding is executed by changing h_7 to 1 and changing l_6 to 0.

During the whole process of embedding, the bit map is left unchanged. As a result, the authentication code can be generated directly from the bit map, and used to compare to the $eauc$ which is stored in the two quantization levels.

3. **Proposed Scheme.** Lin et al.s scheme cannot keep high image quality for a watermarked image when the authentication code is more than three. Based on a novel equation proposed by Zhang et al. [8], we modified it appropriately for our proposed scheme. In our new scheme, we only randomly embed the authentication code into the low level or the high level or bitmap. In other words, only one component of AMBTC compression code for each AMBTC-compressed block will be changed with the proposed scheme. Thus, the image quality of reconstructed watermarked images will be improved comparing with previous schemes. More importantly, even if the authentication code is

more than three, the image quality will also stay high and the range of the authentication code in our scheme is between the decimal number zero and six.

3.1. **Authentication Code Embedding Procedure.** In the proposed scheme, The authentication code (auc) is generated by a pseudo random number generator (PRNG) with a seed, The sender and the receiver know this seed in advance.And the seed is the only side information associated with our proposed scheme. It means that the sender and the receiver can generate the same number rv for each block if they use the same seed. Here, the length of the seed is 8 bits in our scheme and we know that the scheme is more secure with increasing the length of the seed.The processing of generating the authentication code (auc) is shown in the following equation:

$$auc = rv \, mod \, 7 \tag{1}$$

where rv is the random value. The authentication code auc is in the range between zero and six according to the rules of the function we used in [8]. Once authentication auc code is generated, the authentication code embedding procedure begins as follows:

Firstly, the original image is compressed by AMBTC as mentioned above. There is a trio (l, h, BM) for each AMBTC-compressed block. The flowchart for the embedding procedure is shown in Fig. 1 and it is described as follows:
Input: The trio of AMBTC-compressed image block T, the original image, authentication seed
Output: The trio of watermarked AMBTC-compressed image block T'
Step 1: Generate authentication code auc according to Eq. 1.
Step 2: Count the number of ones in BM and denoted as C.
Step 3: Compute the value $f = (1 \times l + 2 \times h + 3 \times C) mod \, 7$.
Step 4: If f equals auc, T is unchanged, $T' = T$; otherwise, the value $s = (auc - f) mod \, 7$ is computed.
Step 5: If s equals 1, set $l = l + 1$.
Step 6: If s equals 2, set $h = h + 1$.
Step 7: If s equals 3 and $(h - l) > 5$, set $l = l + 1$ or $h = h + 1$ depends which modification can avoid s equals 3 or 4 in the next round. Go to Step 3. If s equals 3 but $(h - l) \leq 5$, set $C = C + 1$ and find the pixel which is most close to the average value of the current processing block and its corresponding bit value in BM is 0 in the the original image, and then set such corresponding bit value from 0 to 1 in BM.
Step 8: If s equals 4 and $(h - l) > 5$, set $l = l + 1$ or $h = h + 1$ depends which modification can avoid s equals 3 or 4 in the next round. Go to Step 3. If s equals 4 but $(h - l) \leq 5$, set $C = C - 1$ and find the pixel which is most close to the average value of the current processing block and its corresponding bit value in BM is 1 in the original image, and then set such corresponding bit value 1 to 0 in BM.
Step 9: If s equals 5, set $h = h - 1$.
Step 10: If s equals 6, set $l = l - 1$.

Note that if the value of $l = 255$, $h = 255$ or $C = 16$, and they need to be increased by 1 based on the above procedure. To avoid overflow, they will be decreased by 1 and execute the procedure again instead of increasing them. In contrast, if the value of $l = 0$, $h = 0$ or $C = 0$ and they need to be decreased by 1 based on above procedure. To avoid underflow, they will be increased by 1 and we execute the procedure again instead of decreasing them.

Based on this procedure for a block, the AMBTC-compressed image can be dealt with by repeating this procedure. A watermarking embedding example is provided for $rv = 18$. Consider the block trio is (173, 175, 1101110110000000). According to Eq. 1,

the authentication code auc is computed as 4. We can also count the number of ones in this BM and get the result that $C = 7$. The function f is calculated as Step 3 in authentication code embedding procedure, $f = (1 \times 173 + 2 \times 175 + 3 \times 7) mod\, 7 = 5$. Here f does not equal auc, so we compute s according to Eq. 3, $s = (4 - 5) mod 7 = 6$. So we embed the authentication code by changing 173 to 172. The result for the watermarked compressed trio is (172, 175, 1101110110000000).

In addition, we use an example to explain the method to deal with the problem that the value is saturated or empty. Suppose the block trio is (255, 0, 1111100000000000) and the $rv = 16$, then we can compute $f = 1$ and $auc = 2$. And s can be calculated as 1 further. So we need to change 255 to 256, however, that is not allowed because each pixel value of an image is ranged between 0 and 255. Here we need to change 255 to 254 and execute the procedure again. We can get the new $f = 0$, and $s = 2$ by the new block trio (254, 0, 1111100000000000) and change its corresponding h value from 0 to 1 according to Step 6. Finally, the watermarked compressed trio is (254, 1, 1111100000000000). Analogously, if it is necessary to change h value from 0 to -1, it is also not allowed because each pixel value is ranged between 0 and 255. Therefore, its corresponding value is changed from 0 to 1 and execute the procedure again.

3.2. **Tamper Detection Procedure.** The flowchart for tamper detection is shown in Fig. 2. Tamper detection detects whether the received AMBTC-compressed image has been tampered. The proposed tamper detection procedure is described as follows:
Input: The trio of watermarked AMBTC-compressed image block T' of watermarked image , authentication seed
Output: Tamper detection result of the watermarked image
Step 1: Generate authentication code auc according to Eq. 1.
Step 2: Count the number of ones in BM and denoted as C.
Step 3: If there are more than two kinds of values in the block, mark this block erroneous.
Step 4: Extract authentication code $eauc$ from the block trio by the function $f = (1 \times l + 2 \times h + 3 \times C) mod\, 7$, we take the less value as l and take another one as h in this block, then we assign $eauc = f$.
Step 5: Compare $eauc$ with auc. If they are not equal, mark this block erroneous and complete the detection for this block; otherwise, mark it valid.
Step 6: If all blocks of watermarked image have been proceeded, output the tamper detection result of watermarked image. Otherwise, proceed the next block.

As an example, if the watermarked AMBTC-compressed image block T' =(175,175,172, 175,175,175,172,175,175,172,172,172,172,172,172,172).As we showed above, the $rv = 18$ that generated by the pseudo random number generator (PRNG) with a seed and the seed is the side information. Firstly, we count the number of ones in BM as 7. Then we compute $f = (1 \times 172 + 2 \times 175 + 3 \times 7) mod 7 = 4$, and the $eauc = f = 4$. Finally, we can get $auc = 4$ and compare with $eauc$, and see that they are equal. So this block is valid. It is noted that Step 3 in our tamper detection method mentioned above is a quick judement, if malicious attackers modify the reconstructed image directly, there will be more than two kinds of values in a block. So it will be detected easily. Nevertheless, some malicious attacks may occur in the spatial domain of the reconstructed image of watermarked AMBTC compression code. In other words, the tampered reconstructed image performed by AMBTC encoding, then there will be only two quantization levels for each block, and Step 3 does not work. However, even Step 3 does not work, Step 4 in our tamper detection method still can deal with such attack scenario.

We also need to use a refinement mechanism to improve detection accuracy because it might happen that some blocks were tampered, but its generated authentication code

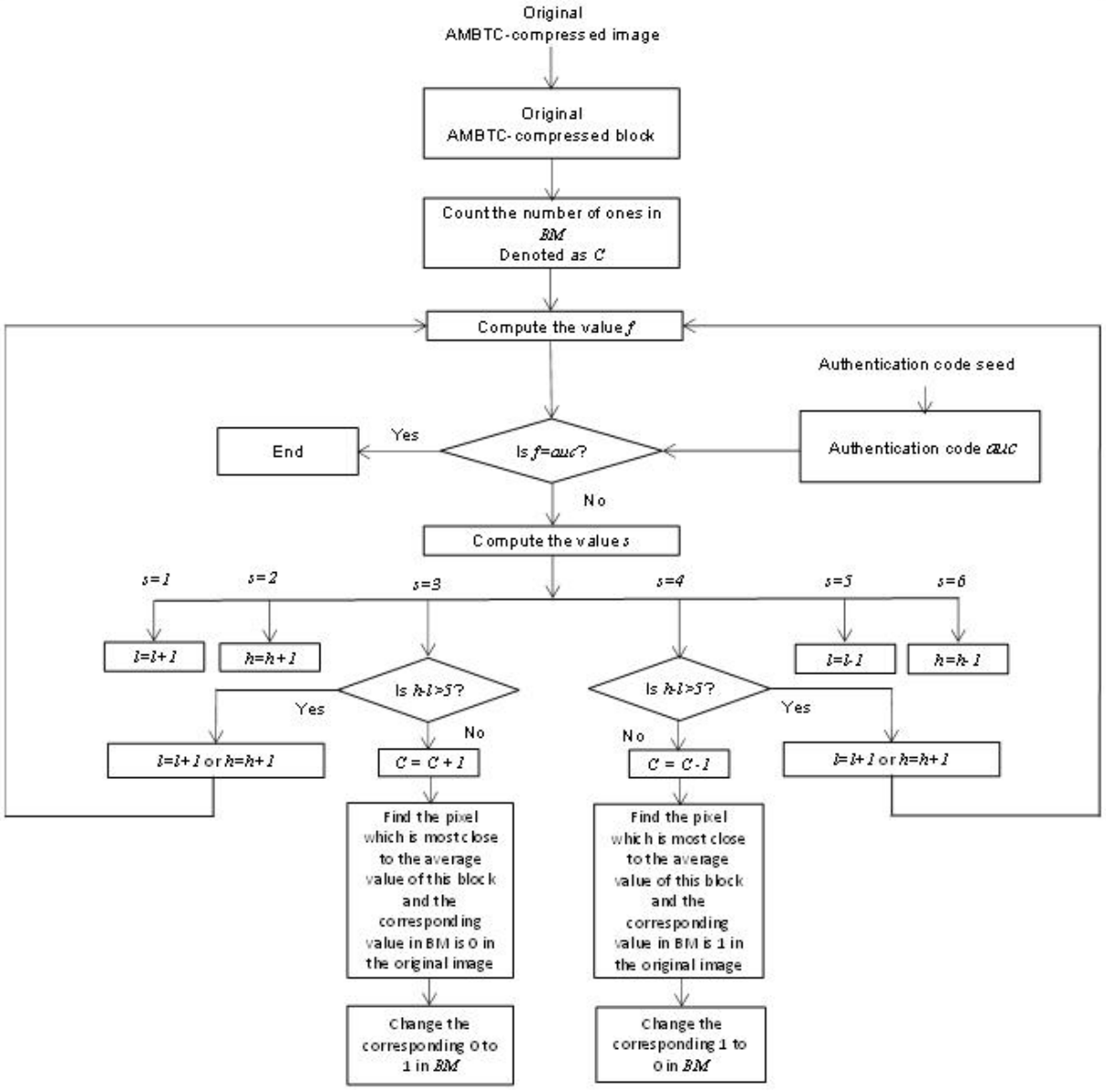

FIGURE 1. Flowchart of embedding procedure

unfortunately equals to the original one. In this case, our designed tamper detection method will not correctly judge it. To avoid it, a refinement mechanism was proposed here. For the refinement mechanism, in its 3×3 block-neighborhood for a block, if they meet one of those four situations shown in Fig. 3, this center block which is the shaded part in Fig. 3. is also marked erroneous.

4. **Experiments and Discussions.** In this section, the superiority of our scheme will be proven by the experimental results. The study used six 512×512 grayscale images as samples and Fig. 4 shows these images, named Lenna, Zelda, Tiffany, Pepper, Girl, and Airplane. The experimental environment is a personal computer running Windows 7 oprating system with an Intel(R) Core(TM) i7-4790 3.60 GHz CPU and 8 GB RAM. Both Lin et al. scheme [16] and our new scheme were implemented by using Dev C++.

The peak signal-to-noise ratio ($PSNR$) is an objective standard for comparing the

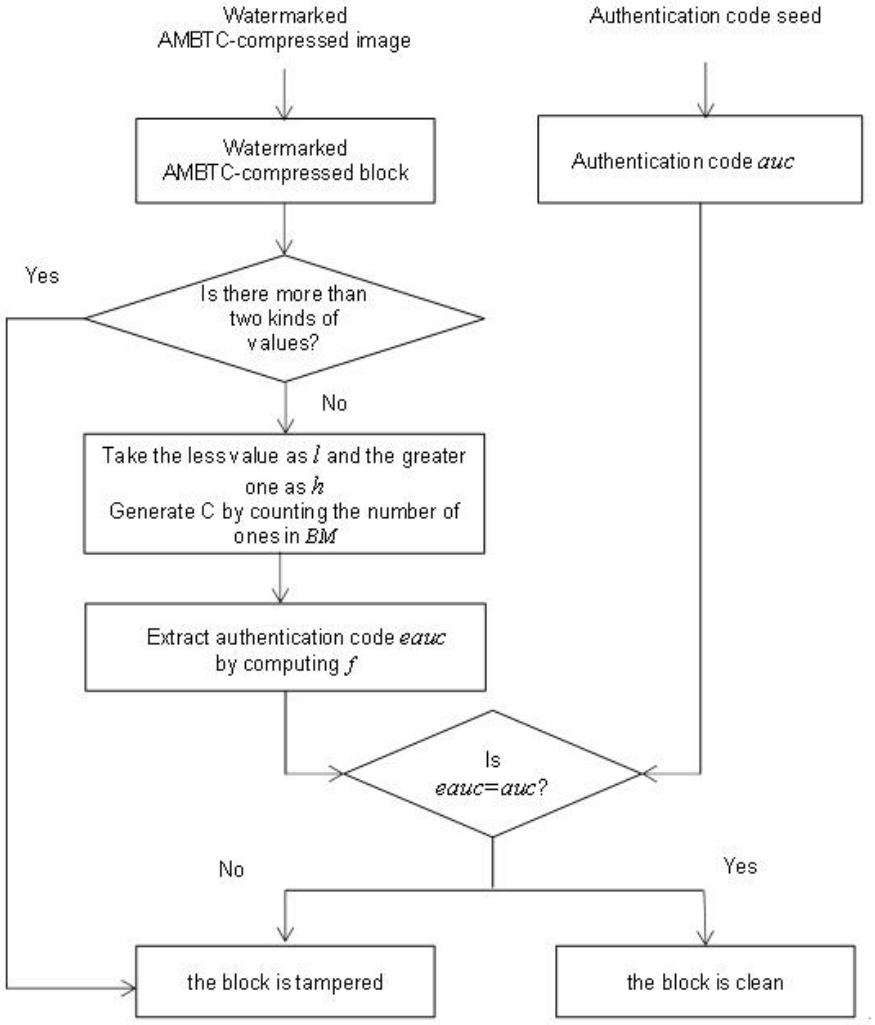

FIGURE 2. Flowchart of tamper detection procedure

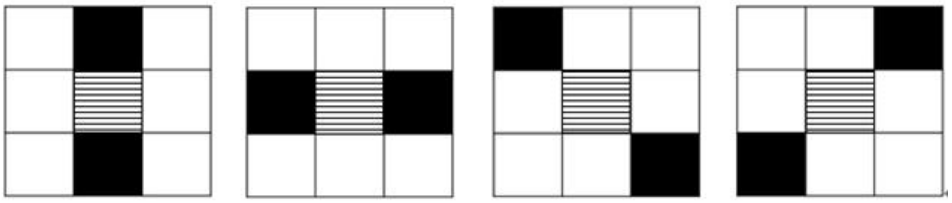

FIGURE 3. The 3×3 block-neighborhood of a valid center block

quality of images:

$$PSNR = 10 \times log_{10} \frac{(2^m - 1)^2}{MSE},$$

m is the number of bits per sample and $m = 8$ in our scheme. The mean squared error (MSE) is introduced as follows:

$$MSE = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} (P_{i,j} - P^*_{i,j})^2,$$

where $P_{i,j}$ is the value of a pixel in the original image, $P^*_{i,j}$denotes the value of a pixel in the watermarked image, and h denotes the height and w width of the image size, respectively.

FIGURE 4. Test images

Firstly, we compressed these six images by AMBTC, and the $PSNR$ values of the reconstructed images with different block sizes are shown in Table 1. Obviously, the image quality decreased with increasing block size.

TABLE 1. $PSNR$s of six AMBTC-compressed images with different block sizes

Block size	2×2	4×4	8×8
Lenna	39.920	33.191	29.904
Zelda	43.817	36.654	32.890
Tiffany	43.167	37.091	33.826
Pepper	40.043	33.175	29.440
Girl	42.141	34.120	30.372
Airplane	39.451	31.953	28.829
Average	41.423	34.352	30.877

In Lin et al.s scheme [16], not only the block size, but also the value of the authentication code can influence the $PSNR$ of a watermarked AMBTC-compressed image and when the number of eb is 1, the authentication code is just one bit in binary form. In other words, the range for the authentication code is between the decimal numbers 0 and 1. They are similar when eb values are 2 and 3, and the corresponding ranges for the authentication code are 0 to 3 and 0 to 7, respectively. In our scheme, the image quality

is not affected by the value of the authentication code, and the authentication code is between the decimal numbers 0 and 6.

Tables 2 to 4 provide comparisons of our scheme and Lin et al.s scheme. First of all, when the authentication code is larger than the decimal number three, the $PSNR$s of our proposed scheme is much better than Lin et al.s scheme although the range of *auc* in our scheme is less than Lin et al.s scheme. The $PSNR$s are 1.45 dB, 0.54 dB, 0.23 dB higher for different block sizes when the *auc* is greater than 3. In addition, after embedding an *auc* which ranges from 0 to 6 by our proposed scheme, the result is even higher than embedding an *auc* that ranges from 0 to 1 in Lin et al.s scheme, except for some specific cases such as when the block size is 2×2. Even under those specific cases, the $PSNR$ results in our scheme are close.

Normally, one of the two quantization levels is just increased or decreased by 1 after embedding the *auc* into the original image block. This also means that we just change the last bit of either l or h in binary formats. It also might change the bit map if a block is smooth, which is determined by whether the difference between l and h in a block is less than 5. In general, Lin et al.s scheme needs to change the last three bits of both l and h. Therefore, our scheme shows excellent performance in image quality.

Table 2. Comparisons for six test images with Lin et al.s scheme [16] and our scheme for the block size 2×2

Ramge of *auc*	0-1[16]	0-3[16]	0-7[16]	Proposed (0-6)
Lenna	39.530	39.308	38.167	39.492
Zelda	42.662	42.154	39.753	42.800
Tiffany	42.834	42.427	41.643	42.329
Pepper	39.596	39.376	38.138	39.607
Girl	41.693	41.385	40.130	41.514
Airplane	39.226	39.054	38.314	39.103
Average	40.924	40.617	39.358	40.808

Table 3. Comparisons for six test images with Lin et al.s scheme [16] and our scheme for the block size 4×4

Ramge of *auc*	0-1[16]	0-3[16]	0-7[16]	Proposed (0-6)
Lenna	33.099	33.070	32.652	33.131
Zelda	36.390	36.330	35.262	36.519
Tiffany	36.931	36.855	36.520	36.885
Pepper	33.075	33.044	32.611	33.114
Girl	34.030	33.990	33.610	34.010
Airplane	31.908	31.885	31.690	31.911
Average	34.239	34.196	33.724	34.262

Finally, the experimental results of tamper detection procedure is shown in Fig. 5 to Fig. 8. Here, we just consider the situation which the block size is 4×4. We tamper the test image named Lenna by embedding a difference image named Rose via photoshop shown in Fig. 5. Of course, the test image was watermarked by *auc* previously. Fig. 6 lists the difference image in pixel and block. In our experiments, we only consider that if the malicious attackers modify the reconstructed image directly and we call it as the first scenario. It is easy to detect the tampered areas which are shown in Fig. 7(a). From

TABLE 4. Comparisons for six test images with Lin et al.s scheme [16] and our scheme for the block size 8×8

Ramge of *auc*	0-1[16]	0-3[16]	0-7[16]	Proposed (0-6)
Lenna	29.857	29.845	29.636	29.878
Zelda	32.776	32.743	32.249	32.836
Tiffany	33.782	33.746	33.563	33.764
Pepper	29.397	29.385	29.187	29.417
Girl	30.330	30.315	30.244	30.330
Airplane	28.805	28.794	28.690	28.809
Average	30.825	30.805	30.595	30.839

FIGURE 5. The tamper samples

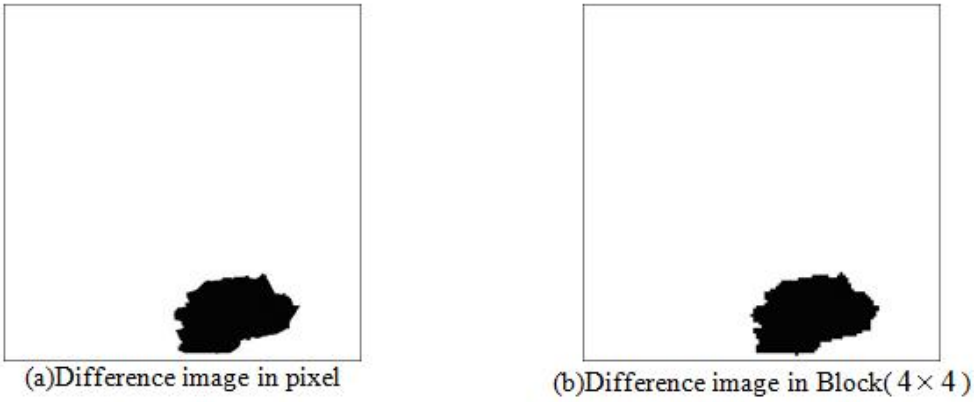

FIGURE 6. Difference image

Fig. 7(a), we can see the accuracy of our tamper detection is high enough even without processed by the refinement mechanism. However, to improve the detection ability further, the ultimate result of tamper detection is listed in Fig. 7(b). By coincidence or misoperation in tampering the watermarked image, some pixels are detected in error shown in Fig. 7(c).

Certainly, the other situation that is the tampered reconstructed image is also performed by AMBTC encoding after embedding a difference image. The result is shown in Fig. 8 and we call this as the second scenario. It is clear that the refinement mechnism is powful and it can increase the accuracy so much. Note that if the accuracy is improved,

then the number of pixels in error image will be decreased.

We also compare the tamper detection rate with Lin et al.s scheme [16] in Table 5, because we use the photoshop to tamper the watermarked image, we cannot tamper two images extremely accurately, they are different in tampered pixels. But the detection rate can prove that the tamper detection accuracy of our scheme is reasonable.

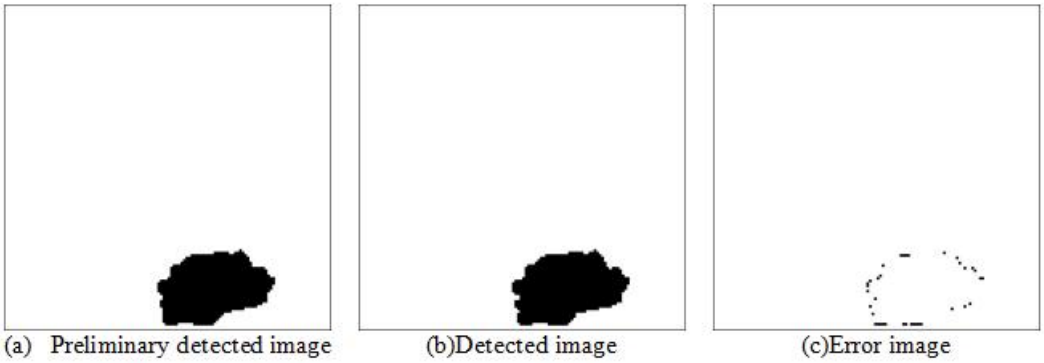

FIGURE 7. (a) detected image without refinement mechanism under the first scneario, (b) detected image with refinement mechanism under the first scneario, (c) error image under the first scnearioo

TABLE 5. Comparisons for tamper detection rate with Lin et al.s scheme [16]

Schemes	Tampered pixels	Error pixels	Detection Rate
Lin et al.s[16]	14256	224	98.43%
Proposed	14848	592	96.01%

5. **Conclusions.** In this paper, we proposed a novel scheme to achieve image authentication of an AMBTC-compressed image. The authentication code was embedded into one element of the trio for the block, or rather, normally, we just modify one bit of the trio in a binary format which may be the last bit of a high level, the last bit of low level, or one bit of a bit map. A comparison of our proposed scheme proves that it outperforms the scheme by Lin et al. The seed of a pseudo random number generator ($PRNG$) was secretly negotiated by a sender and receiver and it is the only side information associated with our proposed scheme. Only a receiver with the correct seed can successfully detect the watermarked image. As some pixels were detected in error, our future work will aim to decrease these errors to as small as possible and further improve the detection ability.

Acknowledgment. The work was supported by the National Natural Science Foundation of China (No. 61173188),the Research Fund for the Doctoral Program of Higher Education (No. 20133401110004), the science and technology project of Anhui Province (No. 1401b042015).

REFERENCES

[1] M. Rabbani and P. W. Jones, Digital Image Compression Techniques, *SPIE*, 10.1117/3.34917, 1991.
[2] Int. Telecommunication Union, CCITT Recommendation T.81,Information Technology, Digital compression and coding of continuous-tone still images-requirements and guidelines, 1992.
[3] E. J. Delp and O. R. Mitchell, Image compression using block truncation coding, *IEEE Trans. Commun*, vol. 27, no. 9, pp. 1335-1342, 1979.

FIGURE 8. (a) detected image without refinement mechanism under the second scenario, (b) error image without refinement mechanism under the second scenario, (c) detected image with refinement mechanism under the second scenario, (d) error image with refinement mechanism under the second scenario

[4] M. D. Lema and O. R. Mitchell, Absolute moment block truncation coding and its application to color image, *IEEE Trans. Commun*, vol. 32, no. 10, pp. 1148-1157, 1984.
[5] F. Bartolini, A. Tefas, M. Bami, I. Pitas, Pitas, Image authentication techniques for surveillance applications, *Proceedings of the IEEE*, vol. 89, no. 10, pp. 14031418, 2001.
[6] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, *IEEE Transactions on Communications*, vol. 28, pp. 8495, 1980.
[7] C. Y. Lin and S. F. Chang, A robust image authentication method distinguish JPEG compression from malicious manipulation, *IEEE Trans. Circ. Syst. Video Tech*, vol. 11, no. 2, pp. 153-168, 2001.
[8] X. Zhang and S. Wang, Efficient Steganographic Embedding by Exploiting Modification Direction, *IEEE Communications Letters*, vol. 10, no. 11, November 2006.
[9] T. Y. Lee and S. D. Lin, Dual watermark for image tamper detection and recovery, *Pattern Recognit*, vol. 41, no. 11, pp. 3497-3506, 2008.
[10] F. Ahmed and M. Y. Siyal, A secure and robust hash-based scheme for image authentication, *Signal Processing*, vol. 90, pp. 1456-1470, 2010.
[11] J. C. Chuang and Y. C. Hu, An adaptive image authentication scheme for vector quantization compressed image, *J. Vis. Commun. Image Represent*, vol. 22, no. 5, pp. 440-449, 2011.
[12] X. Qi and X. Xin, A quantization-based semi-fragile watermarking scheme for image content authentication, *J. Vis. Commun. Image Represent*, vol. 22, no. 2, pp. 187-200, 2011.
[13] C. Li, Y. Wang, B. Ma, and Z. Zhang, Tamper detection and selfrecovery of biometric images using salient regionbased authentication watermarking scheme, *Comp. Stand. Inter*, vol. 34, pp. 367379, 2012.

[14] T. Matsuo and K. Kaoru, On parallel hash functions based on block-ciphers, *Proc. of the IEICE transactions on fundamentals of electronics, communications and computer sciences*, pp. 67-74, 2004.
[15] Y. C. Hu, C. C. Lo, W. L. Chen, and C. H. Wen, Joint image coding and image authentication based on absolute moment block truncation coding, *Journal of Electronic Imaging*, vol. 22, no. 013012, 2013.
[16] C.C. Lin, Y. Huang, W.L. Tai, A high-quality image authentication scheme for AMBTC-compressed images, *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 8, no. 12, pp. 4588-4603, 2014.
[17] C. C. Lo, Y. C. Hu, A novel reversible image authentication scheme for digital images, *Signal Processing*, vol. 98, pp. 174-185, 2014.
[18] M. Goljan, J. Fridrich, and R. Du, Distortion-free data embedding for images, *Proc. of the 4th international workshop on information hiding*, pp. 27-41, 2001.
[19] M. Schneider and S. F.Chang, A robust content based digital signature for image authentication, *Proc. of the IEEE international conference on image processing*, pp. 227-230, 1996.
[20] C.S. Chan, C.C. Chang, An efficient image authentication method based on Hamming code, *Pattern Recognition*, vol. 40, pp. 681691, 2007.
[21] P. W. Wong and N. Memon, Secret and public key image watermarking schemes for image authentication and ownership verification, *IEEE Trans. Image Process*, vol. 10, no. 10, pp. 1593-1601, 2001.
[22] C. Y. Lin and S. F. Chang, A robust image authentication method distinguish JPEG compression from malicious manipulation, *IEEE Trans. Circ. Syst. Video Tech*, vol. 11, no. 2, pp. 153-168, 2001.
[23] X. P. Zhang, Z. X. Qian, Y. L. Ren, G. R. Feng, Watermarking with flexible self-recovery quality based on compressive sensing and compositive reconstruction, *IEEE Trans. Inf. Forensics Secur*, vol. 6, no. 4, pp. 1223-1232, 2011.
[24] B. Schneier. Applied Cryptography: Protocols, Algorithms and Source Code in C. 20th anniversary edition, John Wiley and Sons, Inc, USA, Mar. 2015.
[25] C. S. Shieh, H. C. Huang, T. Y. Chen, and J. S. Pan, BTC-Immunized Watermarking Using Local Homogeneity for Pixel Selection, *Proc. Sixth Int'l Conf. Knowledge-Based Intelligent Information and Engineering Systems*, pp. 422-426, 2002.
[26] Zhi-Hui Wang, Ying-Hsuan Huang, Chin-Chen Chang and Hai-Rui Yang, Reversible Data Hiding for High Quality Using Secret Data Transformation Strategy, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 4, pp. 629-638, July 2015.
[27] H. C. Huang and W. C. Fang, Authenticity Preservation with Histogram-Based Reversible Data Hiding and Quadtree Concepts, *Sensors*, vol. 11, no. 10, pp. 9717- 9731, Oct. 2011.
[28] Wan-Li Lyu, Chin-Chen Chang and Feng Wang, Color PNG Image Authentication Scheme Based on Rehashing and Secret Sharing Method, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 3, pp. 523-533, May 2015.