A High-Quality Image Authentication Scheme for AMBTC-compressed Images

Chia-Chen Lin[1], Yuehong Huang[2], and Wei-Liang Tai[3,*]
[1] Department of Computer Science and Information Management
Providence University, Taichung, Taiwan
[e-mail: ally.cclin@gmail.com]
[2] Institute of Computer Science and Engineering
National Chiao Tung University, Hsinchu, Taiwan
[e-mail: fulva.hyh@gmail.com]
[3] Department of Information Communications
Chinese Culture University, Taipei, Taiwan
[e-mail: tai.wei.liang@gmail.com]
*Corresponding author: Wei-Liang Tai

Received July 17, 2014; revised October 26, 2014; accepted November 9, 2014; published December 31, 2014

Abstract

In this paper, we present a high-quality image authentication scheme based on absolute moment block truncation coding. In the proposed scheme, we use the parity of the bitmap (BM) to generate the authentication code for each compressed image block. Data hiding is used to authenticate whether the content has been altered or not. For image authentication, we embed the authentication code to quantization levels of each image block compressed by absolute moment block truncation coding (AMBTC) which will be altered when the host image is manipulated. The embedding position is generated by a pseudo-random number generator for security concerned. Besides, to improve the detection ability we use a hierarchical structure to ensure the accuracy of tamper localization. A watermarked image can be precisely inspected whether it has been tampered intentionally or incautiously by checking the extracted watermark. Experimental results demonstrated that the proposed scheme achieved high-quality embedded images and good detection accuracy, with stable performance and high expansibility. Performance comparisons with other block-based data hiding schemes are provided to demonstrate the superiority of the proposed scheme.

Keywords: Image authentication, absolute moment block truncation coding (AMBTC), data hiding, tamper detection; security

This research was supported in part under grant number MOST 103-2632-E-126-001-MY3 from Ministry of Science and Technology, Taiwan.

http://dx.doi.org/10.3837/tiis.2014.12.020

1. Introduction

With the rapid development of information technology, the Internet today is used to transmit massive digital images for various kinds of users. In order to speed up the file-transfer rate and decrease image storage space, digital images are always stored in a compressed format based on lossless compression [1] or lossy compression [1]. Lossless compression, does not lose any of the information from the original images, for example, run-length encoding (RLE), entropy encoding, chain codes, and so on. In contrast, lossy compression is subject to loss of some information from the original images in order to achieve a higher compression ratio than lossless compression. As a result, lossy compression is often used to compress general-purpose digital images.

Transform coding [1], block truncation coding (BTC) [1][2][3][4], and vector quantization (VQ) [1] are all typical lossy compression techniques. Among these techniques, BTC proposed by Delp and Ritcell [2] has low computational complexity and excellent reconstructed image quality. Later, Lema and Mitchell [3] redesigned the BTC algorithm by modifying the calculation method of two quantization levels of each AMBTC-compressed block to gain improved reconstructed image quality. This new scheme is called absolute moment block truncation coding (AMBTC) for both grayscale and color image compression.

Data transmitted over the Internet can be easily tampered or copied by malicious users. Image integrity protection has increasingly gained attention due to the rapid development of multimedia and the low security of the Internet. Many solutions have been proposed to ensure digital image authenticity, such as conventional cryptography, digital signatures based on image content, and fragile and semi-fragile watermarking. They could be divided into two groups depending on the service they provide. The first group is formed by strict authentication [5] in which it does not tolerate any manipulation of image data. The second group of applications is frequently addressed as selective authentication [5], which allow for some changes to image data from compression, different filtering algorithms, etc. In practice, however, it is necessary to compress an image to save either storage space or minimize transmission bandwidth/time. Therefore, strict image authentication solutions are less desirable in many practical circumstances.

There are two subgroups within strict authentication services, conventional cryptography based solutions [5][14][15] and fragile watermarking based solutions [5][16][17][18][19][20][21][22]. In general, image authentication solutions included in the conventional cryptography group compute a message authentication code using a hash function that can include message-digest algorithm 5 (MD5) or others. Hash computing for image pixels, rows or columns generates separate hash digests. Next, these hash digests are further encrypted by public key and then appended to the original image. When an image carrying encrypted hash digests is to be verified, receivers first decrypt and then compare the extracted hash digests and the calculated hash digests from the received image. If there are any differences, the image is determined as tampered or vice versa.

For fragile watermarking, there are three main processes for algorithms. First, watermark data are calculated from a set of pixels or other data of an image. Second, the calculated data are embedded into the original image, for example, as another set of pixels. Third, hidden data is extracted when the image is to be authenticated. During the watermark generation procedure, user specified keys are used for encryption and they must be known by senders and receivers in advance. Authenticity of an image can thus be verified. Generally, fragile watermarking algorithms have better performance than algorithms based on conventional cryptography.

Selective authentication service can be grouped into two subgroups, semi-fragile watermarking based solutions [5][23][24][25] and digital-signature based solutions [5][26]. Semi-fragile watermarking solutions aim to insert a watermark into the original image, and differs from fragile watermarking techniques, where the falsification of the protected image can be detected even when the image has undergone some specific image processing operations, i.e., compression, different filtering algorithms, etc. A signature based on image content solutions generally has four main phases. First, high-level characteristics are extracted from the original image. Second, a hash function is used to reduce the size of these characteristics and generate a hash digest. Third, a signature algorithm is used to sign the hash digest to increase security. Finally, the signature to the original image is inserted or attached to the image. When it is necessary to verify the authenticity of the image, the generated image signature is compared to the extracted one. Both signatures should be calculated by the same algorithm.

In 2001, Wong and Memon [6] proposed secret and public key image watermarking solutions for ownership verification and image authentication. Lin and Chang [7] designed a semi-fragile watermarking method that can be used for lossy compressed JPEG images. A compression-domain method [8] that provides JPEG images dual protection was proposed by Lie et al. in 2006. In 2008, Lee and Lin [9] designed a new watermark method for detecting image tamper and recovering the tampered area of the original image. In 2010, a hash-based authentication method [10] was proposed by Ahmed and Siyal to verify the image. In 2011, two additional schemes were proposed, one an adaptive image authentication scheme [11] introduced by Chung and Hu, and the other a quantization-based semi-fragile watermark [12] introduced by Qi and Xin. The former for VQ compressed image authentication and the latter for image content authentication.

In 2013, Hu et al. [13] proposed a new image authentication scheme to verify the tampered areas of AMBTC-compressed images. Hu et al. used a pseudo-random sequence to generate the authentication data. Users determined the number of embedding bits for each AMBTC-compressed image block. Then the authentication code was embedded into certain positions of a bitmap (BM) of each AMBTC-compressed image block. Hu et al.'s method achieved good detection accuracy, however, the visual quality of the embedded image was a little low. The highest quality was 37.8 dB.

As a result, in this paper we propose a new image authentication scheme for AMBTC-compressed images to obtain higher image visual quality and to maintain the detection capability for a watermarked image. In the proposed scheme, the authentication code is embedded into quantization levels of each AMBTC-compressed image block. The experimental results demonstrated that this new authentication scheme achieves both high detection accuracy and high visual quality for the watermarked images.

The rest of the paper is organized as follows. The absolute moment block truncation coding (AMBTC) is provided in Section 2. Section 3 describes a detailed algorithm of our scheme. Experimental results and analysis are shown in Section 4. Finally, conclusions are concluded in Section 5.

2. AMBTC

Based on the BTC, Lema and Mitchell [3] presented a new technique called absolute moment block truncation coding (AMBTC), which provides an improvement in reconstructed image quality. In AMBTC, an image is first partitioned into n×n blocks. Each block can be treated as a vector $x_1, x_2, \ldots, x_k$ of dimension k, where $k= n\times n$. The mean value $\bar{x}$ for each pixel block is computed as

$$\bar{x} = \frac{1}{k}\sum_{i=1}^{k} x_i$$

The mean value is regarded as a threshold that separates the pixels of a block into two groups. If $x_i < \bar{x}$, then x_i is classified into the first group G_0; otherwise the pixel is classified into the second group G_1. For each block, all the pixels will be stored in a bitmap BM. If x_i is in group G_0, x_i will be stored in the bitmap as value 0; otherwise, x_i will be stored in the bitmap as value 1. Then, for each AMBTC-compressed block, the two quantization levels a and b are calculated as

$$a = \frac{1}{k-m}\sum_{x_i<\bar{x}} x_i ,$$

$$b = \frac{1}{m}\sum_{x_i\geq\bar{x}} x_i ,$$

where m is the number of pixels in group G_1. A compressed trio is then generated, (a, b, BM). The decoding method of AMBTC is very simple, and presented as follows.

$$\begin{cases} r_i = a, & if\ bm_i = 0, \\ r_i = b, & otherwise \end{cases} \quad \text{for } 1 \leq i \leq k$$

where $bm_1, bm_2, \ldots, bm_k$ are the values in BM of each compressed trio, and let $r_1, r_2, \ldots, r_k$ are the values of reconstructed vector of each image block.

3. Proposed Scheme

In Hu et al.'s scheme [13], the authentication code is embedded into the bitmap of each AMBTC-compressed block (a, b, BM). This way, the original image is damaged to a large extent, and the visual quality of reconstructed watermarked images is not so good. To improve image quality, we embed the authentication code into quantization levels of each AMBTC-compressed image block so as to create less distortion of the original image. Note that it is necessary to keep quantization level a smaller than quantization level b even after authentication code embedding. In general, two principles are followed in the embedding operations: (1) never change the bits in two quantization levels that are used to generate sequence *ab*; (2) always choose the bit that causes less influence to the quantization level(s) to obtain the higher image quality. Hence, as shown in **Fig. 1**, there are some operations that must be done before embedding watermarks. Having explained our motivation, we now outline the principle of the proposed data hiding algorithm including position code generation procedure, authentication code generation procedure, embedding procedure, and tamper detection procedure.

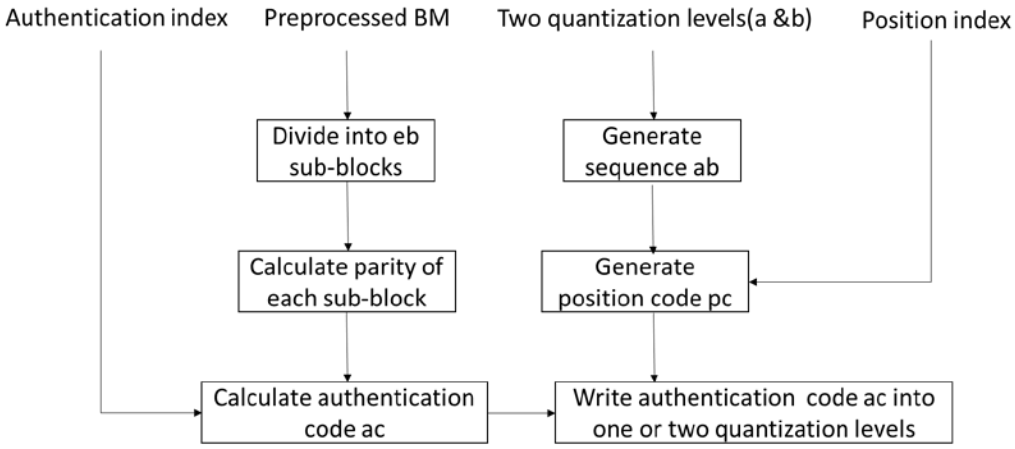

Authentication index
Preprocessed BM
Two quantization levels(a &b)
Position index
Divide into eb sub-blocks
Generate sequence ab
Calculate parity of each sub-block
Generate position code pc
Calculate authentication code ac
Write authentication code ac into one or two quantization levels

Fig. 1. Flowchart of authentication code embedding procedure

3.1 Position Code Generation Procedure

To ensure that the embedding may not change the features of AMBTC, we first use the authentication index and position index to generate the position code where the authentication data is embedded into.

We use a pseudo-random number generator to generate the authentication index and position index for security concerned. The authentication index *ai* and the position index *pi* are computed as

$$ai = rv\_ai \bmod 2^{eb},$$
$$pi = rv\_pi \bmod 8, \tag{1}$$

where *rv_ai* and *rv_pi* are the random values generated by a pseudo-random number generator, and *eb* denotes the number of bits of authentication code.

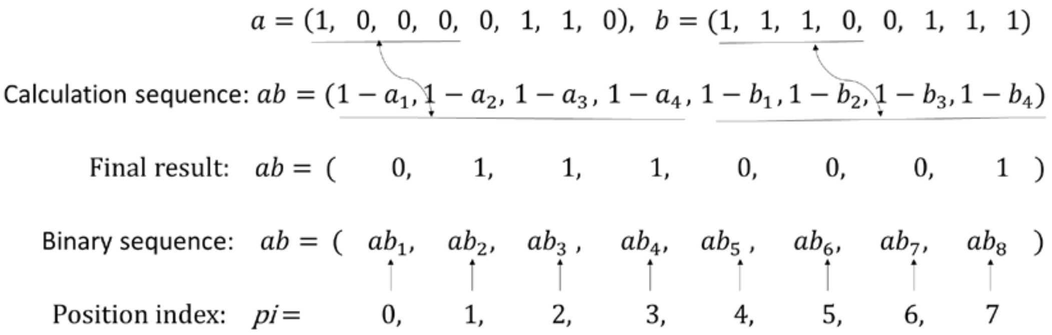
Then, we ues the authentication index and position index to generate the position code *pc* that indicates which bit of quantization level will be used for embedding authentication code *ac*. To ensure the safety of our proposed authentication code embedding procedure, we generate a new sequence called *ab* from two quantization levels to determine the position code *pc*.

Assume that the two quantization levels are transformed to binary formats denoted as $a=(a_1, a_2, \ldots, a_8)$ and $b=(b_1, b_2, \ldots, b_8)$. The *ab* sequence $(ab_1, ab_2, \ldots, ab_8)$ is computed as $ab=\left(\bar{a}_1, \bar{a}_2, \bar{a}_3, \bar{a}_4, \bar{b}_1, \bar{b}_2, \bar{b}_3, \bar{b}_4\right)$derived from two quantization levels, *a* and *b*. The position code *pc* is calculated as

$$\begin{cases} pc = 8 - ab_{pi+1}, & \text{if } eb = 1 \text{ or } 2, \\ pc = 7 - ab_{pi+1}, & \text{if } eb = 3 \text{ or } 4, \end{cases} \tag{2}$$

where *pi* is the position index computed by Eq. 1 and ab_i is an element in sequence *ab* which is chosen by position index *pi*.

The example of generation of sequence *ab* shown in **Fig. 2** is set as *a*=134, *b*=231. The two quantization levels can be presented as two binary sequences: a=10000110, b=11100111. Then the sequence *ab* is obtained as *ab*=01110001. The position code *pc* is computed as **Fig. 3** according to *eb*. If *eb* equals 1 or 2, the possible value of *pc* is 7 or 8; otherwise, if *eb* equals 3 or 4, the possible value of *pc* is 6 or 7. For *eb*=4, we need four bits of two quantization levels to embed watermarks. Therefore, variable position code is used in proposed embedding rather than fix LSB embedding in order to ensure the safety of watermark embedding.



$a = (1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 0)$, $b = (1,\ 1,\ 1,\ 0,\ 0,\ 1,\ 1,\ 1)$

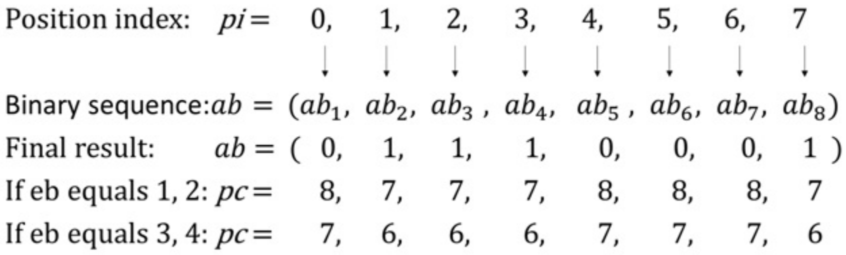
Calculation sequence: $ab = (1 - a_1, 1 - a_2, 1 - a_3, 1 - a_4, 1 - b_1, 1 - b_2, 1 - b_3, 1 - b_4)$

Final result: $ab = (\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 1\ )$

Binary sequence: $ab = (\ ab_1,\ ab_2,\ ab_3,\ ab_4,\ ab_5,\ ab_6,\ ab_7,\ ab_8\ )$

Position index: $pi =$ 0, 1, 2, 3, 4, 5, 6, 7

Fig. 2. Example of generation of sequence *ab*



Position index: $pi =$ 0, 1, 2, 3, 4, 5, 6, 7

Binary sequence: $ab = (ab_1,\ ab_2,\ ab_3,\ ab_4,\ ab_5,\ ab_6,\ ab_7,\ ab_8)$

Final result: $ab = (\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 1\ )$

If eb equals 1, 2: $pc =$ 8, 7, 7, 7, 8, 8, 8, 7

If eb equals 3, 4: $pc =$ 7, 6, 6, 6, 7, 7, 7, 6

Fig. 3. Example of computing position code *pc*

3.2 Authentication Code Generation Procedure

Unlike general watermarking method, instead of external watermarks, we embed features of the image as a watermark into itself for unique authentication. In such self embedding technique the image features are embedded into itself as an authentication data. The features are of course image dependent data and therefore are able to prevent watermarking copy attacks or similar. As a result, the use of image-dependent watermark provides better security as compared to traditional watermarking where a random sequence is used as a watermark for all images [27].

For block-based image authentication, we use the parity of the bitmap (*BM*) to generate the authentication code for each compressed image block. We first divide the bitmap $BM = bm_1, bm_2, \ldots, bm_{n\times n}$ into *eb* sub-blocks with a size

$$bsize = \left\lfloor \frac{n \times n}{eb} \right\rfloor.$$

The sub-block $SDBM_i$ can be presented in the following way

$$SDBM_i = [bm_{1+(i-1)*bsize}, bm_{2+(i-1)*bsize}, \ldots, bm_{bsize+(i-1)*bsize}]$$

The parity p_i of $SDBM_i$ can be computed by

$$p_i = \sum_{j=1}^{bsize} bm_{j+(i-1)*bsize} \mod 2$$

Thus, the authentication code *ac* is generated by

$$ac_i = (ai_i + p_i) \bmod 2, \tag{3}$$

where ac_i is the *i*th bit of authentication code *ac*.

3.3 Authentication Code Embedding Procedure

In the proposed scheme, we embed an *eb*–bit watermark consists of the authentication data into each AMBTC compressed image block. Note that the authentication data is used to identify any modification made to the authenticated image. The proposed watermarking procedure is described as follows.

Input: AMBTC-compressed image *I*, authentication seed, position seed.
Output: Watermarked AMBTC-compressed image *I'*.
Step 1. Generate authentication index *ai* and position index *pi* according to Eq. 1.
Step 2. Generate position code *pc* according to Eq. 2.
Step 3. Generate authentication code *ac* according to Eq. 3.
Step 4. If *eb* equals 1, hide ac_1 into quantization levels as

$$\begin{cases} a_{pc} = ac_1, & \text{if } ac_1 = 0, \\ b_{pc} = ac_1, & \text{otherwise.} \end{cases}$$

Step 5. If *eb* equals 2, hide ac_1 and ac_2 into quantization levels as

$$\begin{cases} a_{pc} = ac_1, b_8 = ac_2, & \text{if } ac_1 = 0, \\ b_{pc} = ac_1, a_8 = ac_2, & \text{otherwise.} \end{cases}$$

Step 6. If *eb* equals 3, hide ac_1, ac_2, and ac_3 into quantization levels as

$$\begin{cases} a_{pc} = ac_1, a_{pc+1} = ac_2, b_{pc+1} = ac_3, & \text{if } ac_1 = 0, \\ b_{pc} = ac_1, b_{pc+1} = ac_2, a_{pc+1} = ac_3, & \text{otherwise.} \end{cases}$$

Step 7. If *eb* equals 4, hide ac_1, ac_2, ac_3, and ac_4 into quantization levels as

$$\begin{cases} a_{pc} = ac_1, a_{pc+1} = ac_2, b_7 = ac_3, b_8 = ac_4, & \text{if } ac_1 = 0, \\ b_{pc} = ac_1, b_{pc+1} = ac_2, a_7 = ac_3, a_8 = ac_4, & \text{otherwise.} \end{cases}$$

Step 8. If quantization level *a* is bigger than quantization level *b*,
for *eb* equals 2, adjust quantization levels as

$$\begin{cases} b_{pc} = 1, & \text{if } pc = 7 \text{ and } ac_1 = 0, \\ a_{pc} = 0, & \text{if } pc = 7 \text{ and } ac_1 = 1, \\ b_{pc-1} = 1, a_{pc-1} = 0, & \text{otherwise.} \end{cases}$$

for *eb* equals 3, adjust quantization levels as

$$\begin{cases} b_{pc} = 1, & \text{if } ac_1 = 0, \\ a_{pc} = 0, & \text{otherwise.} \end{cases}$$

for *eb* equals 4, adjust quantization levels as

$$\begin{cases} b_{pc} = 1, & \text{if } pc = 6 \text{ and } ac_1 = 0, \\ a_{pc} = 0, & \text{if } pc = 6 \text{ and } ac_1 = 1, \\ b_{pc-1} = 1, a_{pc-1} = 0, & \text{otherwise.} \end{cases}$$

Step 9. Repeat Steps 1 to 8 until all AMBTC-compressed blocks have been processed.

Based on definitions of AMBTC, the quantization level a is smaller than quantization level b. Note that Step 8 is used to make sure that quantization level a is still lower than quantization level b after embedding the watermark. A watermarking embedding example is provided for eb=2. Suppose that ai = 01, pi = 5, and the block trio is (10000110, 11100111, 0100111010111100). According to Eq.2, the authentication code ac is computed as 00. The position code pc is calculated as pc=8-(1-ab_6)=8. We embed ac_1 into a_{pc} and ac_2 into b_8 since ac_1=0, that is, a_8=0 and b_8=0. Step 8 is skipped since the quantization level a is smaller than quantization level b in this case. The final watermarked compressed trio is (10000110, 11100110, 0100111010111100).

3.4 Tamper Detection Procedure

The flowchart of tamper detection is shown in **Fig. 4**. Tamper detection detects whether the received AMBTC-compressed image has been tampered. The proposed tamper detection procedure is described as follows.

Input: The watermarked AMBTC-compressed image I', authentication seed, position seed.
Output: Tampered part of the image.
Step 1. Generate authentication index ai and position index pi according to Eq. 1.
Step 2. Generate position code pc according to Eq. 2.
Step 3. Generate authentication code ac according to Eq. 3.
Step 4. Extract authentication code eac from quantization levels according to authentication code ac and position code pc.
Step 5. Compare eac with ac. If they are not equal, mark this block erroneous and complete the detection for it; otherwise, mark it valid.
Step 6. Repeat Steps 1 to 5 until all the blocks have been tested.

To improve the detection ability, we use a hierarchical structure to ensure the accuracy of tamper localization. In the level-2 detection, a valid center block is further marked erroneous if there are two erroneous blocks in its 3×3 block-neighborhood shown in **Fig. 5**.

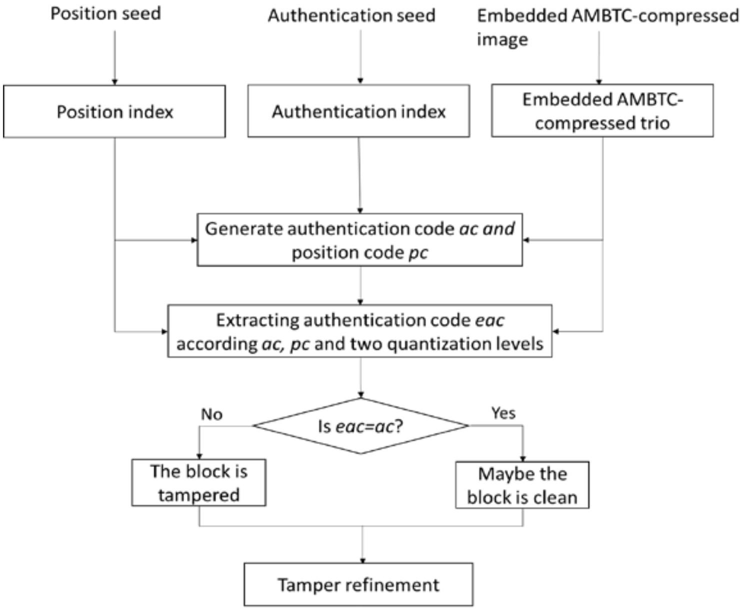

Position seed
Authentication seed
Embedded AMBTC-compressed image
Position index
Authentication index
Embedded AMBTC-compressed trio
Generate authentication code ac and position code pc
Extracting authentication code eac according ac, pc and two quantization levels
No
Yes
Is eac=ac?
The block is tampered
Maybe the block is clean
Tamper refinement

Fig. 4. Flowchart of tamper detection procedure

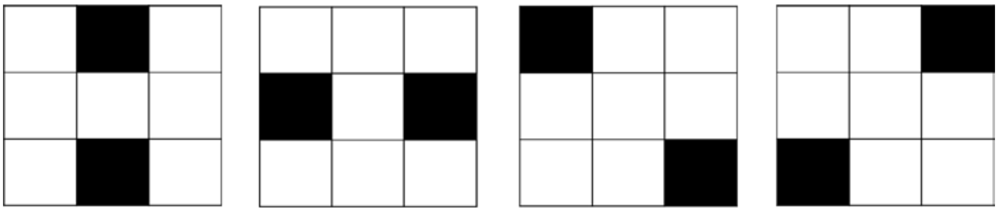


Fig. 5. The 3×3 block-neighborhood of a valid center block

4. Experimental Results

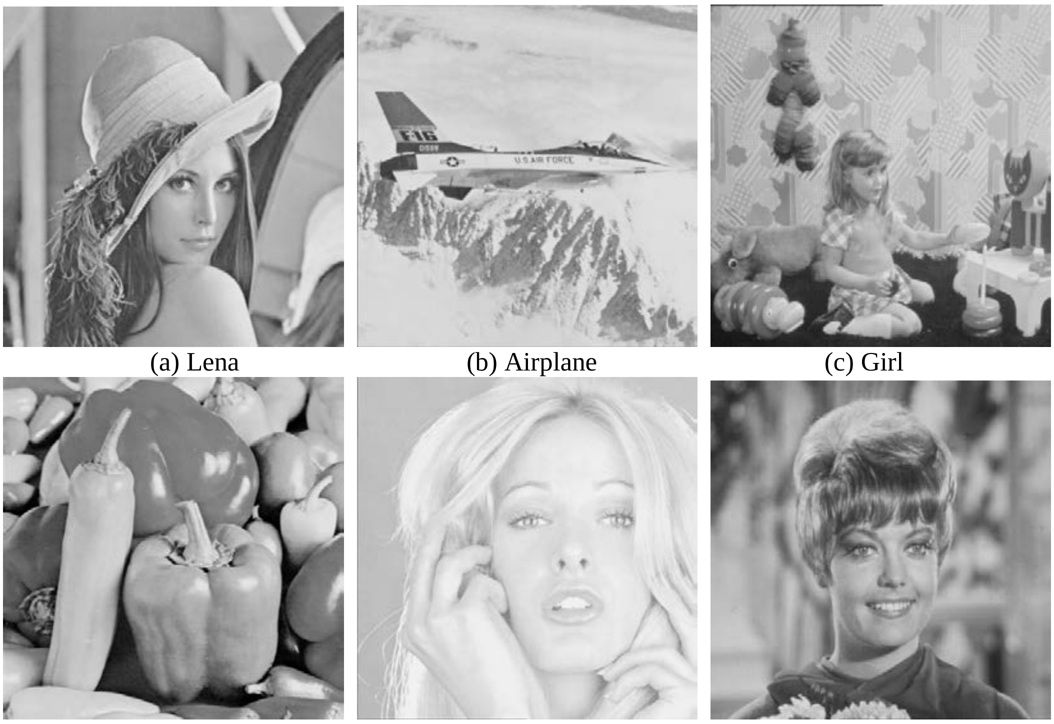
The test images of size 512×512, “Lenna,” “Airplane,” “Girl,” “**Peppers**,” “Tiffany,” and “Zelda,” are shown in **Fig. 6**. To measure the quality of the watermarked images, the peak signal-to-noise ratio (*PSNR*) is introduced as

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right), \tag{4}$$

where the mean squared error (*MSE*) is defined as

$$MSE = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(I_{i,j} - I'_{i,j}\right)^2, \tag{5}$$

where $I_{i,\,j}$ denotes a pixel value of the original image, $I'_{i,\,j}$ denotes a pixel value of the watermarked image, and $H \times W$ is the image size.



(a) Lena (b) Airplane (c) Girl

(d) **Peppers** (e) Tiffany (f) Zelda

Fig. 6. Test images with size 512×512

Table 1 shows the *PSNR* values for the six test images compressed by the AMBTC algorithm with blocks of size 2×2, 4×4, and 8×8. Clearly, the image quality decreases when the block size increases. The average visual qualities are 42.073 dB, 34.969 dB, 31.367 dB when the original image was divided into 2×2, 4×4, and 8×8 blocks, respectively. Note that a higher *PSNR* value resulted in higher image visual quality.

Table 1. Image quality of AMBTC-compressed images with different block sizes

Block size	2×2	4×4	8×8
Lenna	40.683	33.724	30.264
Airplane	40.768	33.286	30.156
Girl	41.935	34.803	31.077
Peppers	41.492	34.103	30.264
Tiffany	43.208	37.082	33.865
Zelda	44.354	36.815	32.577
Average	42.073	34.969	31.367

Tables 2 to **4** compares image quality in *PSNR* for test images delivered by the proposed scheme and Hu et al.'s scheme [13] with block sizes 2×2, 4×4, and 8×8, respectively. The number of bits that are embedded into a block are 1, 2, 3, and 4, respectively. In **Table 2**, our proposed scheme performs better than Hu et al.'s scheme with results that are 3.65 dB, 5.95 dB, 5.16 dB, and 4.92 dB higher, respectively, for different number of embedding bits. The reason is that their scheme embeds the authentication code into the *BM* by modifying the least *eb* distortion bits of *BM*, and also changes the quantization levels of a block. Our scheme, however, only changes the quantization levels of a block. When *eb*=1 and *eb*=2, the maximum distortion for the quantization levels *a* and *b* is no more than two. When *eb*=3 or *eb*=4, maximum distortion for the quantization levels *a* and *b* is no more than six. Therefore, our scheme results in less distortion.

Table 2. Performance comparison for test images with scheme [13] for block size 2×2

eb	1		2		3		4	
Scheme	[13]	Proposed	[13]	Proposed	[13]	Proposed	[13]	Proposed
Lenna	38.11	40.21	35.65	40.03	34.26	38.35	34.04	38.02
Airplane	35.60	40.34	32.93	40.14	31.82	38.59	31.66	38.19
Girl	37.60	41.34	35.15	41.12	33.82	39.10	33.80	38.72
Peppers	38.26	40.91	35.74	40.71	34.35	38.81	34.16	38.45
Tiffany	36.69	42.57	34.17	42.26	33.16	40.12	32.97	39.55
Zelda	40.45	43.29	37.86	42.94	36.61	40.05	36.40	39.60
Average	37.79	41.44	35.25	41.20	34.01	39.17	33.84	38.76

Table 3. Performance comparison for test images with scheme [13] for block size 4×4

eb	1		2		3		4	
Scheme	[13]	Proposed	[13]	Proposed	[13]	Proposed	[13]	Proposed
Lenna	33.70	33.62	33.19	33.58	32.49	33.15	31.92	33.07
Airplane	32.81	33.21	31.83	33.18	30.92	32.82	29.97	32.73
Girl	34.39	34.68	33.66	34.63	32.72	34.09	32.01	33.98
Peppers	34.01	33.99	33.27	33.95	32.45	33.49	31.74	33.39
Tiffany	36.04	36.91	34.83	36.83	33.69	36.12	32.85	35.91
Zelda	36.54	36.61	35.96	36.53	35.16	35.70	34.42	35.56
Average	34.58	34.84	33.79	34.78	32.90	34.23	32.15	34.11

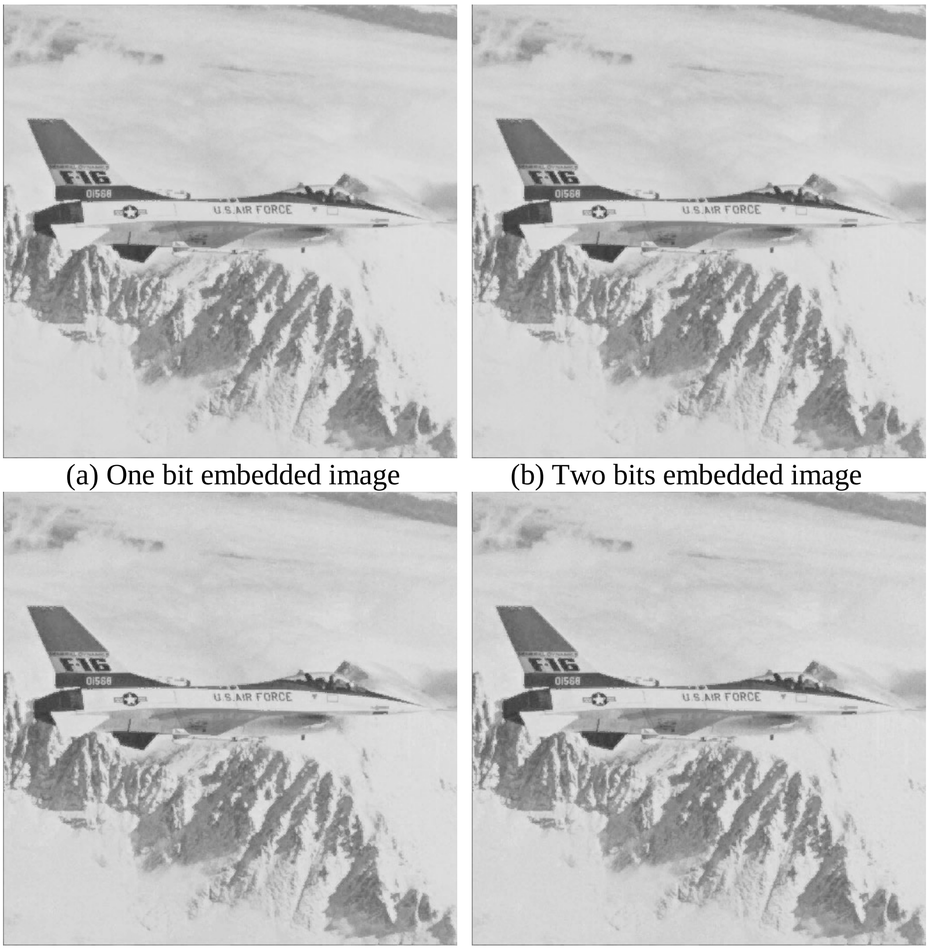
Table 4. Performance comparison for test images with scheme [13] for block size 8×8

eb	1		2		3		4	
Scheme	[13]	Proposed	[13]	Proposed	[13]	Proposed	[13]	Proposed
Lenna	30.33	30.22	30.33	30.20	30.25	30.00	30.15	29.96
Airplane	30.13	30.12	29.95	30.10	29.72	29.94	29.44	29.89
Girl	31.06	31.02	30.94	31.00	30.84	30.76	30.62	30.72
Peppers	30.33	30.22	30.25	30.20	30.10	29.99	29.95	29.95
Tiffany	33.75	33.78	33.42	33.74	33.09	33.40	32.93	33.28
Zelda	32.61	32.49	32.58	32.47	32.47	32.13	32.39	32.07
Average	31.37	31.31	31.25	31.29	31.08	31.04	30.92	30.98

In **Table 3**, the *PSNRs* are still higher than that of the Hu et al. scheme [13] when the size of a block size is 4×4. It was 0.26 dB, 0.9 dB, 1.33 dB, 1.96 dB higher for 1 bit, 2 bit, 3 bit, and 4 bit embedding, respectively. However, in **Table 4** when the number *eb* are 1 and 3, our proposed scheme performs worse than Hu et al.'s scheme. The losses of our scheme are 0.06 dB, and 0.04 dB, respectively. It is obvious that if the size of block increases, the superiority of the *PSNR* in our scheme decreases. This is because the larger the block size, the bigger the bitmap of a block. Whereas, in the meanwhile the quantization levels are still two. Under this condition, if one changes bit information of the bitmap, the loss of image information is significant, and will be shared with the other 63 pixels. However, if you change the quantization levels of an image block, the loss of information is not that significant, but 64 or 32 pixels may have the same loss. Therefore, if the block is larger, the advantage of our scheme is smaller.

As shown in **Tables 2** to **4**, when the number of bits of the authentication code increases, the visual quality of the watermarked image decreases. The visual quality that can be achieved depends on the nature of the image itself. It is also very apparent that the visual quality decreases fast with increased compressed block size. When the size of block is 2×2 or 4×4, our scheme has better *PSNR* values. However, when the block size is increased to 8×8 our *PSNR* value is a little less than that of Hu et al.'s scheme. As such, different authentication schemes may best suited to different block sizes.

Fig. 7 shows the visual impacts of watermarked "Airplane" images at various real hiding capacities for block size 4×4. In general, the watermarked image hardly can be distinguished from the original image. **Fig. 8** compares the visual impacts of the zoomed-in watermarked "Airplane" image for *eb*=4 delivered by the proposed scheme and Hu et al.'s scheme [13]. Clearly, Hu et al.'s scheme destroys the smoothness in many points around edges. The reasion is that Hu et al. scheme transforms a bit in the bitmap from 0 to 1 or 1 to 0 that results in the smooth pixels may become very distinct in the reconstructed image. Usually, the blocks containing the edges may have two quantization levels with a large difference, and sometimes the difference between quantization level *a* and *b* can be more than 100. With this condition, the value of a reconstructed pixel according the transformed bit will turn from quantization level *a* to quantization level *b* or from *b* to *a*. As a result, this pixel may become very distinct in the reconstructed image since it is different from the other pixels around it. Our scheme only changes the value of quantization levels, and thus does not suffer from the same problem.



(a) One bit embedded image (b) Two bits embedded image

(c) Three bits embedded image (d) Four bits embedded image

Fig. 7. Watermarked “Airplane” image

In our experiment, the watermarked image is tampered with by a small cherry image. **Fig. 9** shows the tamper detection result. In our proposed scheme, when embedding an one-bit authentication code into each image block, about half of all tampered blocks can be detected only during the level-1 detection procedure, and only a few blocks around the edges of a tamper image fail to be detected after level-2 detection. From **Fig. 10**, when the bit number of an embedding authentication code is larger, more tampered blocks can be detected after level-2 detection, that is, the failed detected blocks are fewer. Our proposed scheme embeds the authentication code into quantization levels instead of *BM* to achieve good detection accuracy. In our proposed scheme, when embedding a one-bit authentication code into each image block, about half of all tampered blocks can be detected only during the raw detection procedure, and only a few blocks around the edges of a tamper image fail to be detected after tamper refinement. When embedding a two-bit authentication code into each image block, about 3/4 of all tampered blocks can be detected only during the raw detection procedure and the failed tampered blocks are fewer. If the bits of an authentication code is three, about 7/8 of tampered blocks can be detected only during the raw detection procedure. If the bits of an authentication code is four, about 15/16 of the tampered blocks can be detected only during the raw detection procedure, and the failed detected blocks are the fewest.

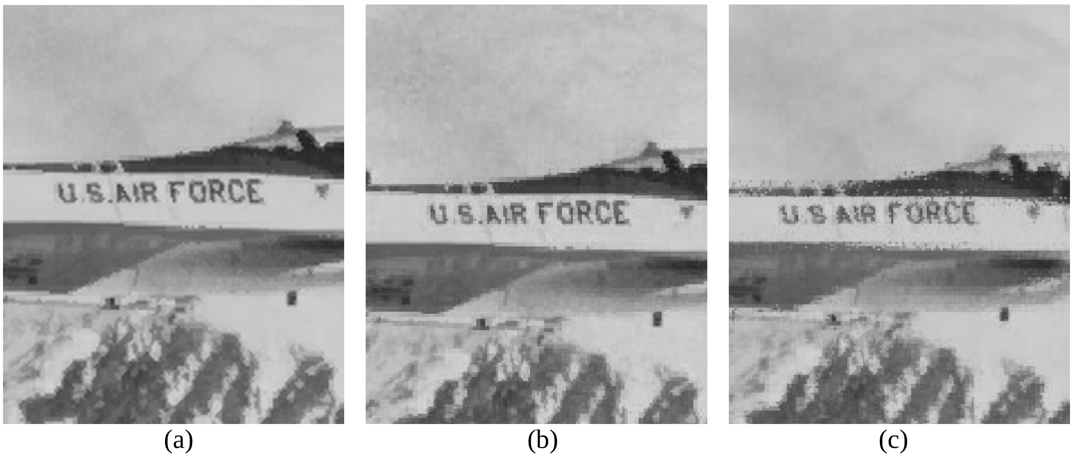
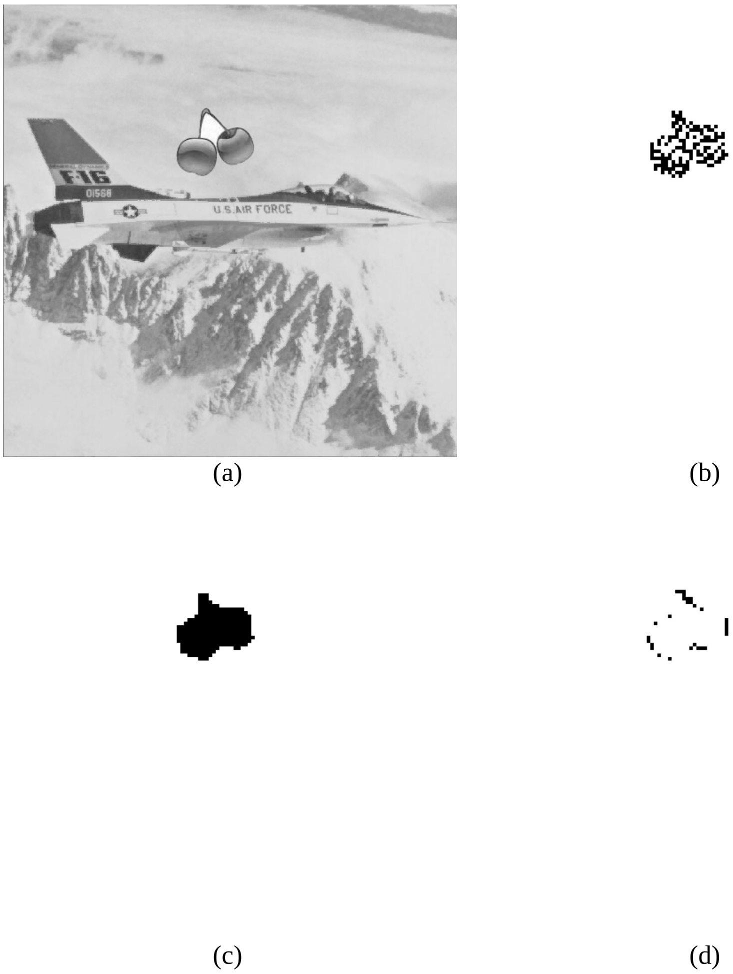
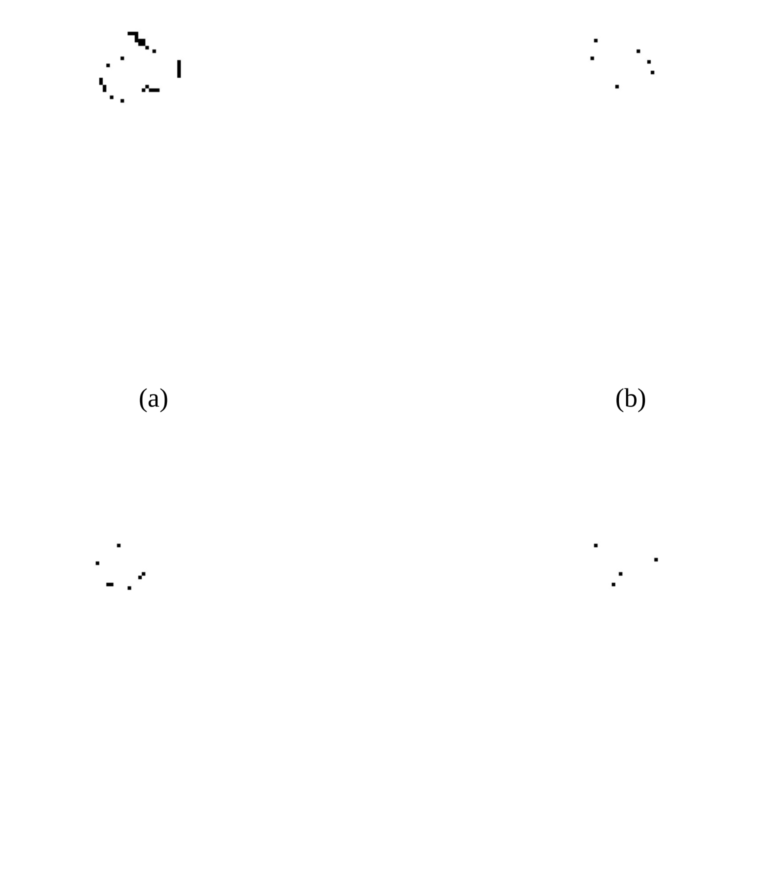

(a) (b) (c)

Fig. 8. Performance comparison for the zoomed-in “Airplane” image with scheme [13] for *eb*=4; (a) AMBTC-compressed image, (b) watermarked image using our proposed scheme, and (c) watermarked image using Hu et al.’s scheme


(a) (b)

(c) (d)

Fig. 9. Tamper detection results for *eb*=1; (a) tampered “Airplane”, (b) level-1 detection, (c) level-2 detection, and (d) Failed detected blocks



(a) (b)

(c) (d)

Fig. 10. Failed detected results for various hiding capacity; (a) *eb*=1, (b) *eb*=2, (c) *eb*=3, and (d) *eb*=4

5. Conclusion

In this paper, we present a high-quality image authentication scheme to protect the security of AMBTC-compressed images. The authentication code derived from the *BM* information is protected by the random numbers. Our proposed scheme embeds the authentication code into quantization levels instead of *BM* to obtain higher visual quality in the embedded image. Our scheme has better visual quality in comparison to Hu et al., which results in some abrupt points appearing around the edges of an embedded image. In the future, we will try to propose an adaptive image authentication scheme for AMBTC compressed images, which can offer high quality for AMBTC compressed images with different block sizes.

References

[1] M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*, SPIE, 1991. Article (CrossRef Link).

[2] E. J. Delp and O. R. Mitchell, “Image compression using block truncation coding,” *IEEE Transactions on Communications*, vol. 27, no. 9, pp. 1335-1342, Sept. 1979.

Article (CrossRef Link).

[3] M. D. Lema and O. R. Mitchell, “Absolute moment block truncationcoding and its application to color image,” *IEEE Transactions on Communications*, vol. 32, no. 10, pp. 1148-1157, Oct. 1984. Article (CrossRef Link).

[4] P. Fränti, O. Nevalainen, and T. Kaukoranta, “Compression of digital images by block truncation coding: a survey,” *The Computer Journal*, vol. 37, no. 4, pp. 308-332, 1994. Article (CrossRef Link).

[5] A. Haouzia and R. Noumeir, “Methods for image authentication: a survey,” *Multimedia Tools and Applications*, vol. 39, no. 1, pp. 1-46, Aug. 2008. Article (CrossRef Link).

[6] P. W. Wong and N. Memon, “Secret and public key image watermarking schemes for image authentication and ownership verification,” *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1593-1601, Oct. 2001. Article (CrossRef Link).

[7] C. Y. Lin and S. F. Chang, “A robust image authentication method distinguish JPEG compression from malicious manipulation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 153-168, Feb. 2001. Article (CrossRef Link).

[8] W. N. Lie, G. S. Lin, and S. L. Chen, “Dual protection of JPEG images based on informed embedding and two-stage watermark extraction techniques,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 3, pp. 330-341, Sept. 2006. Article (CrossRef Link).

[9] T. Y. Lee and S. D. Lin, “Dual watermark for image tamper detection and recovery,” *Pattern Recognition*, vol. 41, no. 11, pp. 3497-3506, Nov. 2008. Article (CrossRef Link).

[10] F. Ahmed and M. Y. Siyal, “A secure and robust hash-based scheme for image authentication,” *Signal Processing*, vol. 90, no. 5, pp. 1456-1470, 2010. Article (CrossRef Link).

[11] J. C. Chuang and Y. C. Hu, “An adaptive image authentication scheme for vector quantization compressed image,” *Journal of Visual Communication and Image Representation*, vol. 22, no. 5, pp. 440-449, Jul. 2011. Article (CrossRef Link).

[12] X. Qi and X. Xin, “A quantization-based semi-fragile watermarking scheme for image content authentication,” *Journal of Visual Communication and Image Representation*, vol. 22, no. 2, pp. 187-200, Feb. 2011. Article (CrossRef Link).

[13] 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. 1, 2013. Article (CrossRef Link).

[14] A. Harry, *VDM Specification of The MD4 Message Digest Algorithm*, National Physical Laboratory, Teddington, 1992. Article (CrossRef Link).

[15] T. Matsuo and K. Kurosawa, “On parallel hash functions based on block-ciphers,” *IEICE transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E87-A, no. 1, pp 67-74, 2004. Article (CrossRef Link).

[16] J. Fridrich, M. Goljan, and R. Du, “Invertible authentication,” in *Proc. of SPIE 4314, Security and Watermarking of Multimedia Contents III*, San Jose, CA, Jan. 2001. Article (CrossRef Link).

[17] M. Goljan, J. Fridrich, and R. Du, “Distortion-free data embedding for images,” in *Proc. of 4th Int. Workshop on Information Hiding*, pp 27-41, 2001. Article (CrossRef Link).

[18] A. van Leest, M. van der Veen, and F. Bruekers, “Reversible image watermarking,” in *Proc. of 2003 Int. Conf. on Image Processing*, vol. 3, pp 731-734, Step. 2003. Article (CrossRef Link).

[19] H. Guo, Y. Li, A. Liu, and S. Jajodi, “A fragile watermarking scheme for detecting malicious modifications of database relations,” *Information Sciences*, vol. 176, no. 10, pp. 1350-1378, 2006. Article (CrossRef Link).

[20] C. Qin, C. C. Chang, and P. Y. Chen, “Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism,” *Signal Processing*, vol. 92, no. 4, pp. 1137-1150, 2012. Article (CrossRef Link).

[21] X. Zhang, S. Wang, and G. Feng, “Fragile watermarking scheme with extensive content restoration capability,” in *Proc. of the 8th Int. Workshop on Digital Watermarking*, pp. 268-278, 2009. Article (CrossRef Link).

[22] Anthony T. S. Ho, X. Zhu, J. Shen, and P. Marziliano, “ Fragile watermarking based on encoding of the zeros of the transform,” *IEEE Transactions on Information Forensic and Security*, vol. 3, no.

3, pp. 567-569, Sept. 2008. Article (CrossRef Link).

[23] J. Fridrich, "Image watermarking for tamper detection," in *Proc. of 1998 Int. Conf. on Image Processing*, vol. 2, pp. 404-408, 1998. Article (CrossRef Link).

[24] J. Fridrich, M. Goljan and A. C. Baldoza, "New fragile authentication watermark for images," in *Proc. of 2000 Int. Conf. on Image Processing*, vol. 1, pp. 446-449, Sept. 2000. Article (CrossRef Link).

[25] G. Friedman, "The trustworthy digital camera: restoring credibility to the photographic image," *IEEE Transactions on Consumer Electronics*, vol. 39, no. 4, pp. 905-910, 1993. Article (CrossRef Link).

[26] M. Schneider and S. F. Chang, "A robust content based digital signature for image authentication," in *Proc. of 1996 Int. Conf. on Image Processing*, vol. 3, pp 227-230, Sept. 1996. Article (CrossRef Link).

[27] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," in *Proc. of 1999 Int. Conf. on Image Processing*, vol. 3, pp. 792-796, Oct. 1999. Article (CrossRef Link).

Chia-Chen Lin (also known as Min-Hui Lin) received her B.S. degree in information management in 1992 from the Tamkang University, Taipei, Taiwan. She received both her M.S. degree in information management in 1994 and Ph.D. degree in information management in 1998 from the National Chiao Tung University, Hsinchu, Taiwan. Dr. Lin served as Visiting Associate Professor at the University of Illinois at Urbana Champaign during the period of August 2006 to July 2007. The visiting scholarship was appointed and sponsored by the Ministry of Education, Taiwan. Dr. Lin is currently a professor of the Department of Computer Science and Information Management, Providence University, Sha-Lu, Taiwan. From August 2008 to August 2009, she was the Associate Dean of Academic Affairs of Providence University. From August 2009 to August 2010, she is the Dean of Research and Development of Providence University. She is also a member of IEEE and a member of ACM. From July 2010 to July 2012, she served Vice Chairman of Tainan Chapter, IEEE Signal Processing Society. Currently, she is the editor of KSII Transactions of Internet and Information Systems, the associate editor of Journal of Journal of Information Hiding and Multimedia Signal Processing and Journal of Electronic Science and Technology. She is also the regional editor of Recent Patents on Computer Science. She has served guest editors for Soft Computing, Journal of Computers and Journal of Electronic Science and Technology. Her research interests include image and signal processing, information hiding, mobile agent, and electronic commerce.

Yuehong Huang received the M.S. degree in computer science and information management from Providence University, Taiwan, in 2014. She is currently pursuing the Ph.D. degree in institute of computer Science and engineering from National Chiao Tung University, Taiwan. Her research interests include image processing and Internet.

Wei-Liang Tai received the M.S. degree in computer science and information engineering from National Chung Cheng University, Taiwan, in 2004 and the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Taiwan, in 2008.

He is currently Assistant Professor, Department of Information Communications, Chinese Culture University. His main interests are in information security and forensics and multimedia signal processing. He is currently an Editor of The Scientific World Journal for the "Signal Processing."