electronics

MDPI

Article

Fragile Watermarking for Tamper Localization and Self-Recovery Based on AMBTC and VQ

Chia-Chen Lin [1,]*, Ting-Lin Lee [2], Ya-Fen Chang [3,]*, Pei-Feng Shiu [4] and Bohan Zhang [5]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan
[2] Department of Asia-Pacific Industrial and Business Management, National University of Kaohsiung, Kaohsiung 811, Taiwan
[3] Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung City 404, Taiwan
[4] Department of Computer Science and Engineering, National Chung Hsing University, Taichung City 40227, Taiwan
[5] Institute of Information Management, National Yang Ming Chiao Tung University, Taipei 10650, Taiwan
* Correspondence: ally.cclin@ncut.edu.tw (C.-C.L.); cyf@nutc.edu.tw (Y.-F.C.); Tel.: +886-4-2392-4505 (ext. 8740) (C.-C.L.); +886-4-2219-6610 (Y.-F.C.)

Abstract: Digital images have unique features that include being both easily transmittable over the Internet and being easy to tamper. With the advancement of digital processing techniques and an increasing number of valuable digital images being transmitted via the Internet, image authentication has been made more crucial than ever. In this paper, we present an image authentication scheme with tamper localization and self-recovery using fragile watermarking. We embed the fragile watermarks consisting of the authentication code and the recovery information onto the image to verify its integrity. The proposed fragile watermarking scheme can authenticate the image without accessing the original image, localizing the modifications as well as verifying the integrity, and even reconstructing the tampered regions. We use an AMBTC compressed code as the authentication code to minimize the distortion introduced by embedding. To reduce the blocking effect that occurs in the reconstructed image, a VQ compressed code is applied instead of the average intensity as the recovery information. Several representative test images and 200 different test images were randomly selected from BOWS to examine the performance of the proposed scheme. Experimental results confirm that the proposed scheme can effectively resist a cutting attack and a copy-paste attack while retaining the high accuracy of tamper localization. The average TPR and average FTP rate were around 97% and 0.12%, respectively, while maintaining the image quality of the watermarked image and restoring the image at up to 48 dB and 39.28 dB, respectively.

Keywords: fragile watermarking; image authentication; AMBTC; VQ

check for updates

Citation: Lin, C.-C.; Lee, T.-L.; Chang, Y.-F.; Shiu, P.-F.; Zhang, B. Fragile Watermarking for Tamper Localization and Self-Recovery Based on AMBTC and VQ. *Electronics* **2023**, *12*, 415. https://doi.org/10.3390/electronics12020415

Academic Editor: Frederic Ros

Received: 13 November 2022
Revised: 25 December 2022
Accepted: 26 December 2022
Published: 13 January 2023

1. Introduction

Digital images that are easy to edit, modify, and exploit can be widely shared and distributed via the Internet. Thanks to powerful image processing techniques, it is increasingly easier for everyone to perfectly edit digital images and create forgeries. Creating perfect forgeries can lead to the theft and misuse of intellectual property. Proving the origin of an image and its integrity is thus essential for an image owner. As a result, image authentication and integrity verification have become important issues in recent years.

In general, the authenticity of digital images can be guaranteed by using digital signatures. Digital signatures employ asymmetric cryptography to establish the authenticity and integrity of a digital image by attaching a hash of the image. One possible drawback of digital signatures is the fact that extra bandwidth is required to transmit the signatures. Moreover, authentication based on digital signatures cannot localize changes nor reconstruct tampered regions in the image, even if integrity protection is provided.

To address the above problem, fragile watermarking schemes have been proposed as a means of verifying image integrity [1]. The visual redundancy of digital images makes it possible to embed invisible fragile watermarks into such images without modifying the essential features of the images. To further enhance the function of fragile watermarking schemes, researchers have designed self-embedding methods, such as that of a fragile watermark concurrently consisting of an authentication code and recovery information. With the hidden authentication code and recovery information, the modifications made to the watermarked image are expected to be localized, and tampered regions can later be restored. Different from robust watermarks, for both fragile watermarking and self-embedding fragile watermarking, embedded watermarks can be easily compromised by any kind of malicious attack [1–3]. In other words, any attempt to alter the image content will also alter the embedded fragile watermark itself, which is thus capable of detecting every change that has occurred to the image [4–22].

In 2011, Lee et al. designed a hierarchical fragile watermark based on VQ index recovery [5]. In their scheme, with the hierarchical strategy and LSB substitution, the average image quality of the watermarked image was around 39.6 dB. In the next year, He et al. [6] presented a self-recovery fragile watermarking scheme using block-neighborhood tamper characterization. They generated nonlinear block mapping to embed the watermark and used an optimized neighborhood characterization method to detect tampering. In the next year, Zhang et al. [7] proposed a self-embedding fragile watermarking scheme. In their scheme, they first generated DCT coefficients for each 2 × 2 block. Next, they embedded the generated fragile watermark into another block according to the block mapping. The experiments confirm that the average PSNR of the watermarked images is around 42.6 dB, and the tampered regions can be successfully localized and exactly recovered for content-only tampering.

In 2014, Lin et al. [8] proposed a high-quality image authentication scheme based on absolute moment block truncation coding (AMBTC). They used the parity of the bitmap to generate the authentication code for authenticating each compressed image block. The proposed hierarchical inspection structure was effective in resisting a collage attack. Unfortunately, Lin et al.'s scheme did not offer a recovery feature. In the same year, Yang et al. designed a fragile watermarking with a recovery function for halftone images [9]. In the next year, Sarreshtedari and M. A. Akhaee [10] compressed the whole image as recovery data and encoded it as a watermark by using Reed-Solomon codes (RS codes). The tampered regions can be restored by the error-correcting RS code. However, they cannot restore more than n-k erasures when using the RS(n, k) code. In 2015, Li et al. designed a reference matrix-based watermark embedding strategy to conceal authentication codes into quantization levels of the compressed images generated by the block truncation coding (BTC) [11]. In 2016, Qin et al. [12] designed self-embedding watermarking based on a reference-data interleaving mechanism and adaptive selection of the embedding mode. In their scheme, the binary bits in the adopted MSB layers are scrambled and individually interleaved with different extension ratios and are then combined with authentication bits to form the watermark bits for LSB embedding. In Qin et al.'s scheme, the average image quality of the recovered image remained at 45.42 dB when the tamper rate was around 12%. In 2017, Cao et al. [13] proposed a self-embedding fragile image watermarking scheme for tamper recovery. They also adopted MSB layers and LSB layers, but they applied a hierarchical recovery mechanism in their scheme. According to the contribution of the image quality, the binary bits in the MSB layers were first scrambled and then individually interleaved with different extension ratios. Later, the interleaved data served as recovery information and were embedded into LSB layers of non-overlapping blocks along with the authentication code. Even with the hierarchical recovery mechanism, the average image quality of the restored images was not significantly improved compared to Qin et al. [12]. In the same year, Qin et al. designed a VQ-based self-embedding fragile watermarking [14]. In their scheme, VQ indices derived from the original image served as recovery information. Later, hash values were computed from the combination of VQ indices, and the original image was computed and embedded into the image itself via LSB substitution. Different from Qin et al. [14], Lin et al. designed a hybrid watermark hiding strategy for compressed images generated by [15]. Although their scheme did not provide a recovery function, the image quality of the watermarked images had been significantly improved compared with existing schemes.

In 2018, Tai and Liao [16] embedded the fragile watermark of one block into another block according to the embedding sequence generated by a chaotic map. To reduce the smooth blocking effect of the recovered images, they used a wavelet transform rather than the average as the recovery data to enhance the image contrast. This method can effectively resist a collage attack and constant-average attack. Hong et al. [17] proposed an efficient authentication scheme for AMBTC compressed images. They protected the AMBTC codes by embedding the authentication codes into the least significant bits (LSBs) of two quantization levels to minimize the embedding distortion. In 2019, Chen et al. [18] proposed a novel authentication scheme for the AMBTC of a compressed image using turtle-shell-based data hiding. Previous AMBTC-based schemes have the problem of having a high quantization level, which is lower or equal to a low quantization level caused by the hiding operation. Thus, they proposed an iterative embedding mechanism to solve the above issues and achieved high tamper detection accuracy. Su et al. [19] presented an authentication scheme based on the matrix encoding for AMBTC-compressed images. The six-bit authentication code was embedded into two sub-bitmaps using matrix encoding. Their scheme offered an improved detection rate in the first hierarchical tampering detection. In 2020, Roy et al. [20] designed a copyright protection mechanism with digital image watermarking. In their scheme, adaptive LSB replacement was adopted to embed the watermark. Moreover, to enhance the robustness of the hidden watermark, the higher bit-planes were also modified instead of only the LSB. In 2021, Hong et al. [21] further improved the visual qualities of marked and recovered images by using matrix encoding and side match techniques. In the same year, Chang et al. used AMBTC compression results to first generate a watermark. Later, they applied the turtle shell data hiding method to conceal the watermark in the original image [22]. Lin et al. [23] presented a pixel pair-wise fragile image watermarking method. They used AMBTC to generate the authentication code and recovery information as watermarks. To reduce overhead information, the bitmap generated by AMBTC was further compressed by Huffman coding.

Although many self-embedding fragile watermarking schemes have been proposed in the last six years, it continues to be a challenge to enhance the tamper detection capacity and image quality of the restored images while maintaining an acceptable visual quality in the watermarked images. In particular, an increasing number of valuable digital images are being transmitted via the Internet and shared through social media platforms. Neither individual users nor companies are willing to compromise the quality of their personal images or digital image productions when they try to adopt integrity protection mechanisms. Thus, in this paper, we present a fragile watermarking scheme as an image assurance tool for integrity protection with tamper localization and self-recovery. Tamper detection performance is related to the size of the hidden authentication code, and recovery performance is related to the size of the recovery information and the correlation between the recovery information and the original image. However, the larger the amount of the hidden authentication code and recovery information, the lower the image quality of the watermarked image is. To maintain the tradeoff between the performance of tamper detection and recovery, and the image quality of the watermarked image, AMBTC compressed codes derived from the original image are used as the authentication code that can detect every possible change that has occurred in an image with a very high probability. To allow the tampered regions of the image to be partially repaired, we utilized vector quantization (VQ) compressed codes as the recovery information. The contributions of this paper are summarized as follows:

- A novel fragile watermarking combining an AMBTC compression method is designed to enhance tamper detection performance. Using the AMBTC compression codes to generate authentication codes, the average TPR, and average FTP rate are around 97% and 0.12%, respectively, which outperforms other existing schemes.
- Utilizing VQ indices as the recovery information, the image quality of the restored image ranges from 31.26 dB to 46.05 dB. Even in the worst case, the PSNR is still above 30 dB.
- With the increased concealment of the authentication codes, our scheme provides better tamper detection performance against a copy-paste attack and cutting attack where different tamper ratios are encountered.
- The experimental results confirm that the proposed fragile watermarking scheme exceeds the performance of most existing work with respect to balancing tamper detection and the image quality of watermarked images and restored images.

Section 2 reviews the basic concepts of AMBTC and VQ that are needed for the fragile watermarking scheme. Section 3 explains the proposed watermarking scheme, including the embedding, detection, and self-recovery algorithms. Experimental results and their analysis appear in Section 4. Finally, the paper is concluded in Section 5.

2. Related Work

This section reviews two image compression techniques that are necessary to generate the proposed fragile watermark. The first is AMBTC which is used to generate the authentication code, while the second technique is VQ, which is used to generate the recovery information.

2.1. AMBTC

Absolute moment block truncation coding (AMBTC) [24] is based on the idea of block truncation coding, whereas it is simpler in the implementation. The pixels in each block, which are quantized into two-level outputs, make the mean value, and the first absolute central moment is preserved in the reconstructed block. AMBTC preserves absolute moments rather than standard moments, resulting in a lower mean squared error (MSE) with the bit rate as low as 2 bpp (bit per pixel).

The image is divided into blocks of 4 × 4 pixels. The mean value $\overline{x}$ of each block x, taken as the one-bit quantizer threshold, is computed as:

$$\overline{x} = \frac{1}{16}\sum\nolimits_{i=1}^{16} x_i \tag{1}$$

where x_i is the ith pixel in the block. A bitmap BM is used to record the thresholding result, which is generated as:

$$bm_i = \begin{cases} 0 & \text{if } x_i < \overline{x} \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where bm_i is the ith bit in the bitmap. Two-level quantization outputs are computed from:

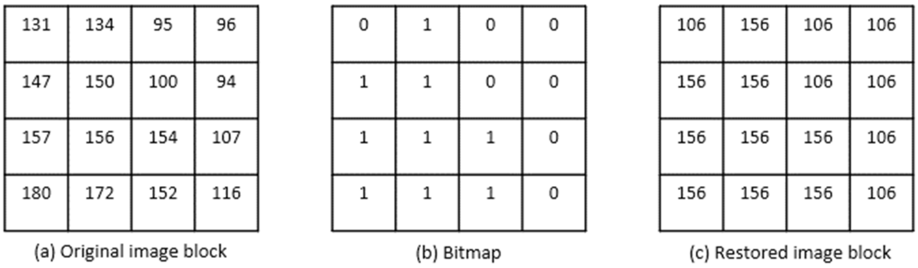
$$\overline{x}_L = \frac{1}{16-q}\sum\nolimits_{x_i<\overline{x}} x_i \tag{3}$$

$$\overline{x}_H = \frac{1}{q}\sum\nolimits_{x_i\geq\overline{x}} x_i \tag{4}$$

where q is the number of pixels whose values are larger than the mean value, $\overline{x}_L$ and $\overline{x}_H$ denotes the lower and higher means of the block, respectively. As a result, each block is encoded as a triple ($\overline{x}_L$, $\overline{x}_H$, BM). For AMBTC decoding, each pixel x'_i in the block can be reconstructed as:

$$x'_i = \begin{cases} \overline{x}_L & \text{if } bm_i = 0 \\ \overline{x}_H & \text{otherwise.} \end{cases} \tag{5}$$

An example demonstrating AMBTC compression is shown in Figure 1. Figure 1a shows an original image with a block sized 4 × 4 pixels, and its corresponding bitmap can be derived based on Equation (2) when the average value of the original image block is computed. According to the bitmap shown in Figure 1b, two quantization levels for the two groups are calculated by Equations (3) and (4). Finally, the restored image block can be constructed based on two quantization levels and a bitmap.



131	134	95	96
147	150	100	94
157	156	154	107
180	172	152	116

(a) Original image block

0	1	0	0
1	1	0	0
1	1	1	0
1	1	1	0

(b) Bitmap

106	156	106	106
156	156	106	106
156	156	156	106
156	156	156	106

(c) Restored image block

Figure 1. Example of AMBTC Compression: (**a**) Original image block, (**b**) Bitmap and (**c**) Restored image block.

2.2. Vector Quantization

Vector quantization (VQ) [25] is a classical quantization method that can be used for image compression. It divides a large set of vectors into groups that have similar vectors using clustering algorithms. Each group is represented by its centroid vector. VQ encodes a vector as the index of its closest centroid and is thus capable of achieving image compression with a low error and bit rate. VQ has been successfully used in a vector-quantized variational autoencoder (VQ-VAE) for the high-quality and large-scale generation of images.

VQ maps k-dimensional vectors in the vector space R^k into a finite set of vectors as $Y = \{y_0, y_1, \ldots, y_{n-1}\}$ and $k = w \times h$, where h is the height of a block and w is the width of a bock. The vector y_i is called a codeword, and the set of all the codewords Y is called a codebook, as shown in Figure 2. Given an input vector, the encoder determines the representative codeword y_i that is the closest in Euclidean distance from it and, thus, encodes the input vector as the index i of the codeword y_i. At the decoder, this index i is used to lookup the codeword y_i from the same codebook Y to reconstruct the vector. Note that the performance of VQ is mostly influenced by the elements of the codebook. Theoretically, VQ can be useful in cases when the decoder has limited information and a fast execution time is required.

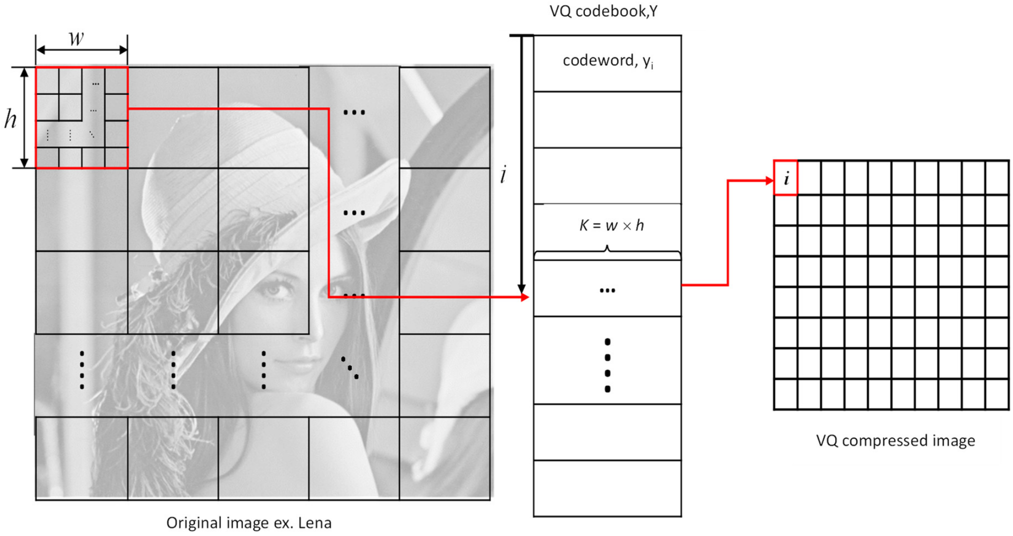


Figure 2. Example of VQ compression.

3. Proposed Method

We propose a fragile watermarking scheme for image authentication with the capabilities of tamper localization and self-recovery. A fragile watermark consisting of an authentication code and recovery information is self-embedded in the image making, thus, is capable of authenticating itself without accessing the original image. To further improve tamper detection performance and recovery performance while not causing significant distortion to the original image, the AMBTC compression codes and VQ indices derived from the original image are served as the authentication code and recovery information, respectively. Note that a general codebook is adopted in our scheme during the generation of VQ indices instead of a unique and pre-trained codebook to eliminate extra transmission costs for sharing the VQ codebook. A detailed description of watermark generation and embedding, and authentication/recovery, are given in Sections 3.1 and 3.2, respectively.

3.1. Watermark Generation and Embedding

The process of the watermark generation and embedding is illustrated in Figure 3. We assume that the original image is an 8-bit grayscale digital image of n pixels, where all possible pixel values are integers in the range [0, 255]. The original image is divided into non-overlapping blocks of 4 × 4 pixels. For each block, we generated an 8-bit authentication code derived from its corresponding AMBTC compression code that would be used to detect any modification made to the image. To generate the AMBTC compression codes for a given image, we first used a random seed γ to generate a random bit stream of length $4n$ bits and then used the bit stream to substitute 4 LSBs of the original image to obtain the preprocessed image. For each block of 4 × 4 pixels in the preprocessed image, we computed the triple $(\overline{x}_L, \overline{x}_H, BM)$ by AMBTC and generated the 8-bit authentication code AC as:

$$\mathrm{AC} = H(BM)\ ||\ H(\overline{x}_H\ ||\overline{x}_L\ ), \tag{6}$$

where $H(.)$ is a 4-bit hash function. $H(BM)$ indicates the 4-bit hash value derived from the bitmap, BM. $H\ (\overline{x}_H\,||\,\overline{x}_L)$ indicates the 4-bit hash value derived from the two quantizers. The recovery information RI of each block is generated by VQ encoding of the original image with a codebook size of 256.

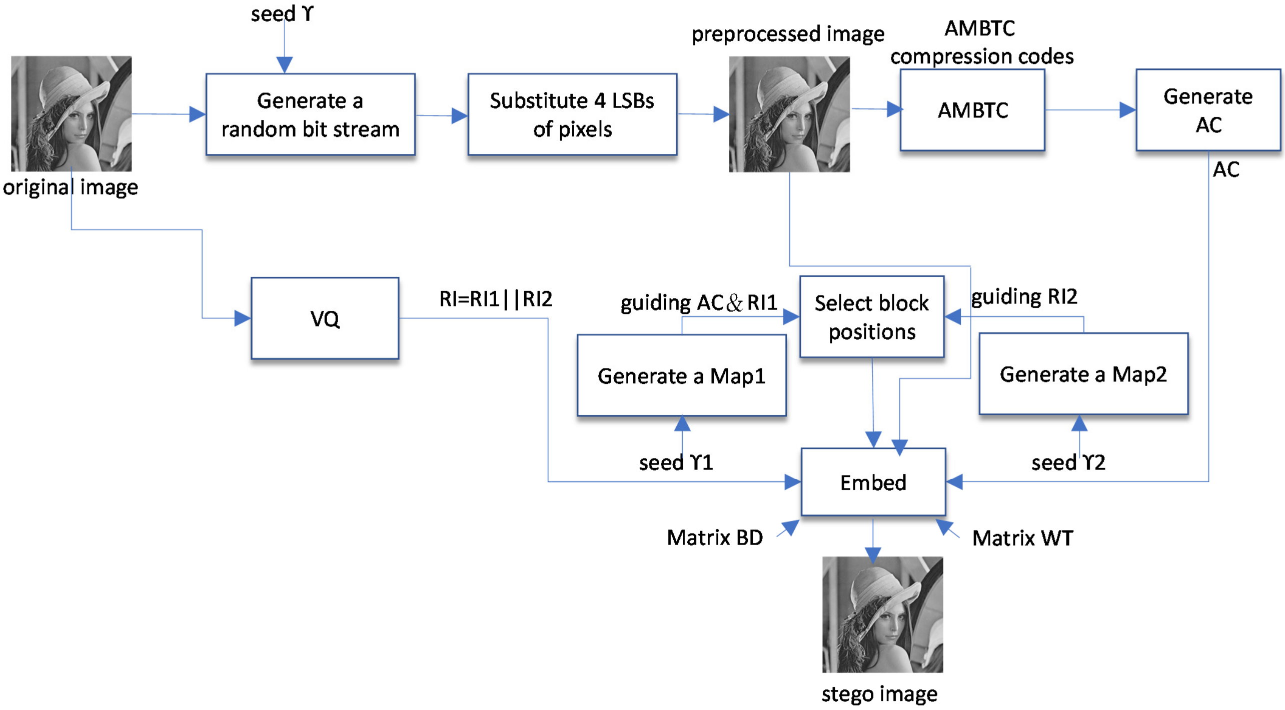


Figure 3. Watermark generation and embedding process.

To make the authentication with tamper localization and self-recovery possible, for each block, we embedded an authentication code AC into it for tamper localization while embedding the recovery information RI for image recovery. It was necessary to break the block-wise independency to resist malicious attacks. Here, we used random seeds $\gamma 1$ and $\gamma 2$ to generate two non-repeating sequences, Map1 and Map2, respectively, for watermark generation. To enhance the tamper detection performance of the hidden authentication code and ensure that the tampered regions could be restored by their recovery information, the AC and RI for a given block were not directly embedded into its block. Instead, RI was separated into two parts, and embedded into two different blocks according to Map1 and Map2, as shown in Figure 4. For each block, we embedded a 24-bit fragile watermark msg composed of the AC and RI1 of its mapping block selected from Map1 and the RI2 of the other mapping block selected from Map2 into it, which could be further divided into eight 3-bit watermarks as:

$$\text{msg} = \{\text{AC}, \text{RI1}, \text{RI2}\} = \{\text{msg1}, \ldots, \text{msg8}\}, \quad (7)$$

where msgi = {ACi, RI1i, RI2i}.

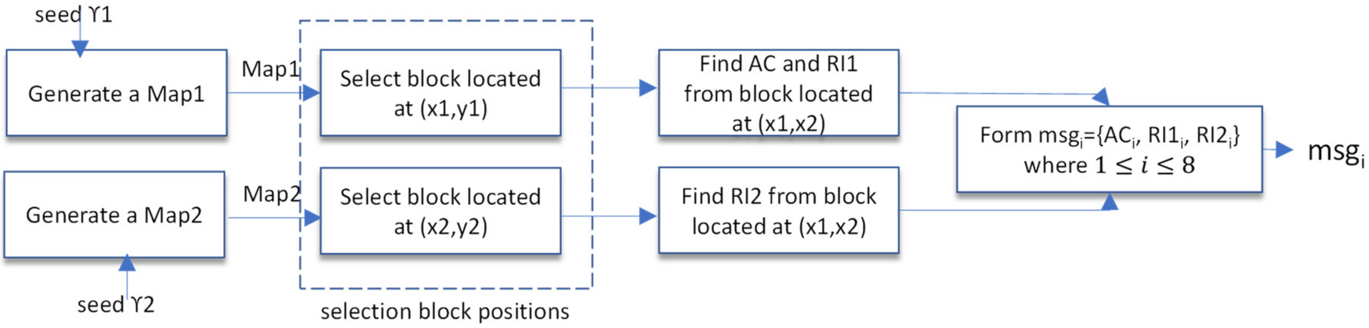

seed Y1
Generate a Map1
Map1
Select block located at (x1,y1)
Find AC and RI1 from block located at (x1,x2)
Form msg$_i$={AC$_i$, RI1$_i$, RI2$_i$} where 1 ≤ *i* ≤ 8
msg$_i$
Generate a Map2
Map2
Select block located at (x2,y2)
Find RI2 from block located at (x1,x2)
seed Y2
selection block positions

Figure 4. Process for selecting embedding block position and generating the hidden message AC and RI.

The watermark embedding operation was then designed so that a 3-bit watermark msg_i could be assigned to a pair of pixels. We further propose a binary digital table (BD) and waves type table (WT) to modify the pair of pixel values to meet their assigned watermark msg_i. Figure 5 shows the binary digital table BD, which is defined as:

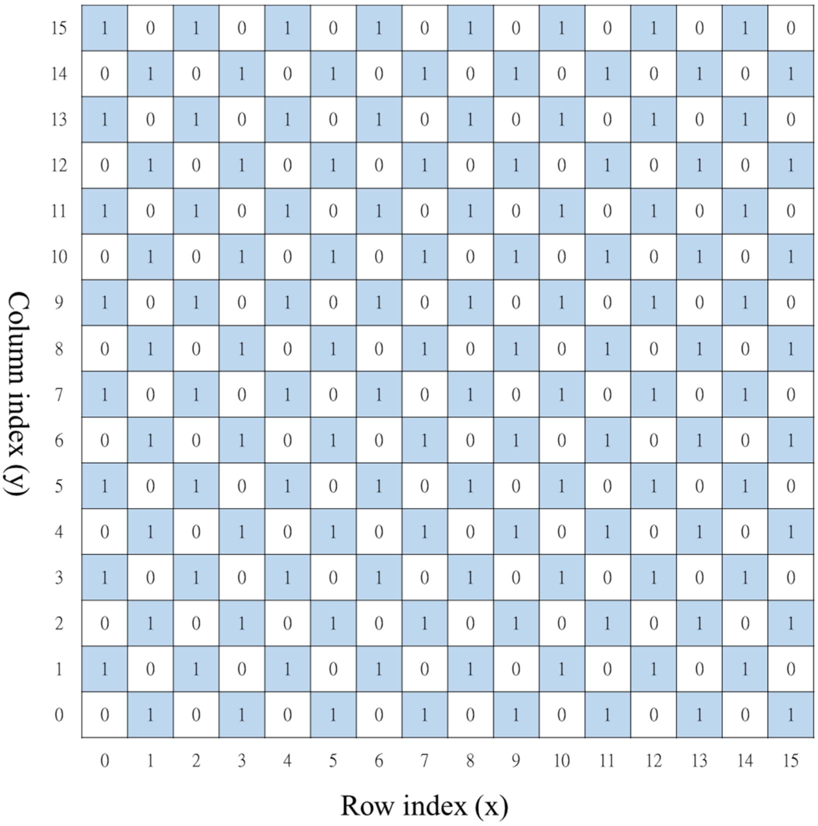
$$\text{BD}_{\text{x, y}} = (\text{x} + \text{y}) \bmod 2, \quad (8)$$

where $\text{BD}_{\text{x, y}}$ is the binary value at the position (x, y). The waves type table WT is illustrated in Figure 6 and is computed by:

$$WT_{x,y} = (x + \lfloor y/2 \rfloor) \bmod 4, \quad (9)$$

where $\text{WT}_{\text{x, y}}$ is the table value at the position (x, y). To see how to embed a 3-bit watermark msgi in a pair of pixels, let us read the next two pixels, P1 and P2, of a block as a pair in raster order. The 4 LSBs of the pair are used to generate the coordinate (x, y) = (P1 mod 16, P2 mod 16). From matrix BD and matrix WT, we found the coordinate (x′, y′) in the clockwise direction from (x, y) such that $\text{BD}_{\text{x}', \text{y}'} = \text{AC}_\text{i}$ and $\text{WT}_{\text{x}', \text{y}'} = 2 \times \text{RI1}_\text{i} + \text{RI2}_\text{i}$. The embedded pair of pixels (P1′, P2′) was then computed by:

$$\begin{cases} \text{P1}' = 16 \times \lfloor \text{P1}/16 \rfloor + x' \\ \text{P2}' = 16 \times \lfloor \text{P2}/16 \rfloor + y' \end{cases} \quad (10)$$

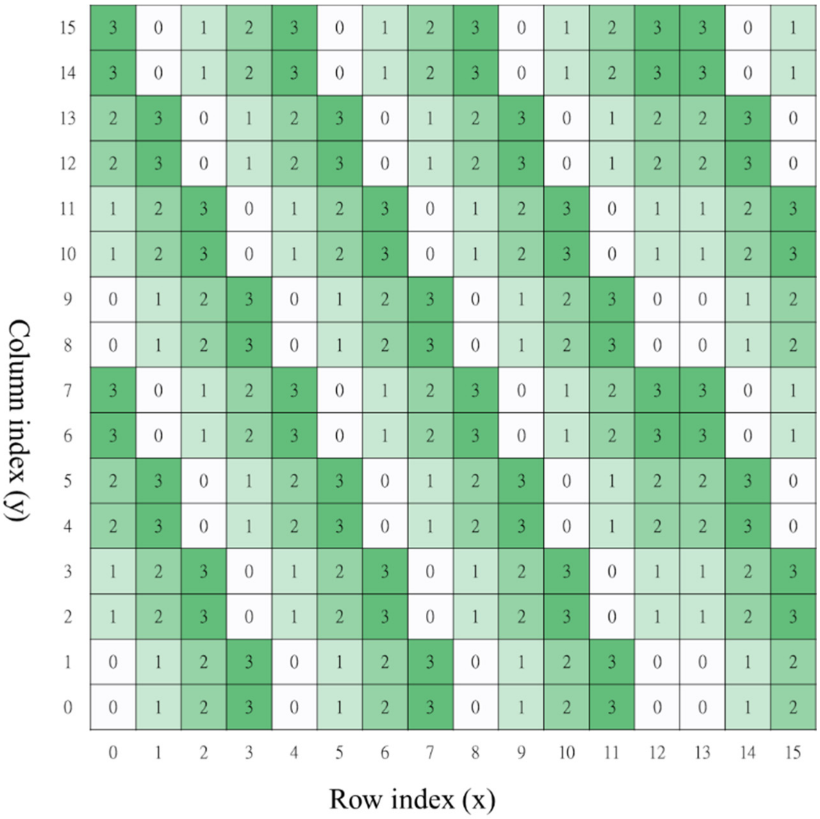


Column index (y)

15	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
14	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
13	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
12	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
11	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
10	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
9	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
8	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
7	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
6	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
5	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Row index (x)

Figure 5. Binary digital table BD.



Column index (y)

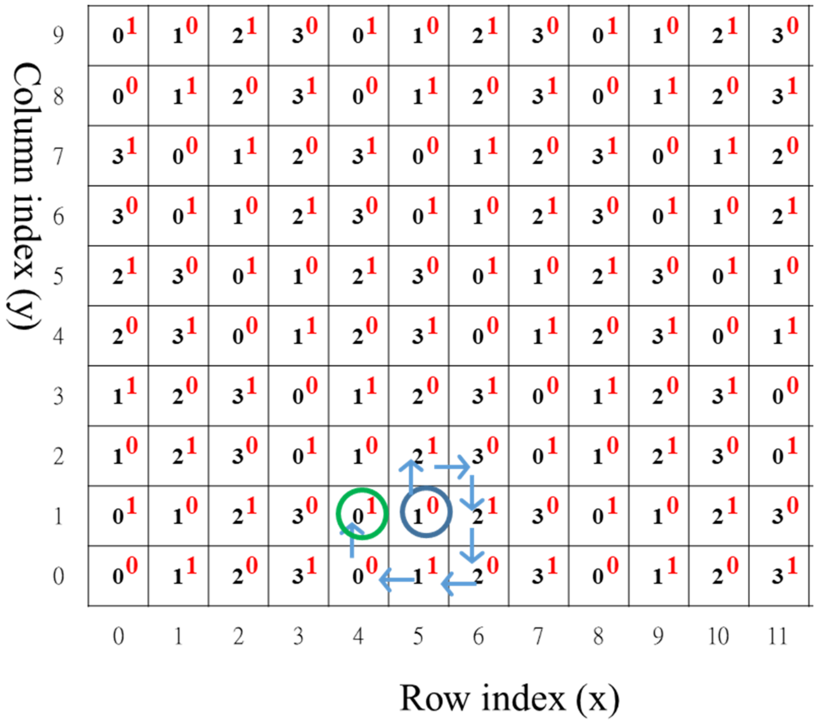
15	3	0	1	2	3	0	1	2	3	0	1	2	3	3	0	1
14	3	0	1	2	3	0	1	2	3	0	1	2	3	3	0	1
13	2	3	0	1	2	3	0	1	2	3	0	1	2	2	3	0
12	2	3	0	1	2	3	0	1	2	3	0	1	2	2	3	0
11	1	2	3	0	1	2	3	0	1	2	3	0	1	1	2	3
10	1	2	3	0	1	2	3	0	1	2	3	0	1	1	2	3
9	0	1	2	3	0	1	2	3	0	1	2	3	0	0	1	2
8	0	1	2	3	0	1	2	3	0	1	2	3	0	0	1	2
7	3	0	1	2	3	0	1	2	3	0	1	2	3	3	0	1
6	3	0	1	2	3	0	1	2	3	0	1	2	3	3	0	1
5	2	3	0	1	2	3	0	1	2	3	0	1	2	2	3	0
4	2	3	0	1	2	3	0	1	2	3	0	1	2	2	3	0
3	1	2	3	0	1	2	3	0	1	2	3	0	1	1	2	3
2	1	2	3	0	1	2	3	0	1	2	3	0	1	1	2	3
1	0	1	2	3	0	1	2	3	0	1	2	3	0	0	1	2
0	0	1	2	3	0	1	2	3	0	1	2	3	0	0	1	2
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Row index (x)

Figure 6. Waves type table WT.

Although the mapping operations are complex, its idea is quite straightforward. Our objective is that pixels in a block only require slight modifications so that the modified pixel pair's coordinate can map to its corresponding AC and RI via matrices BD and WT. In other words, to carry the 24-bit hidden watermark, we divided it into eight segments, and each segment contained only a 3-bit watermark called msg_i, where $1 \leq i \leq 8$ for an embedding block sized 4×4. Two pixels in an embedding block form a pixel pair, and two transformations are defined in Equations (8) and (9). With these two transformation functions, the corresponding 3-bit watermark can generate two different values when mapped to matrices BD and WT, respectively, as shown in Figures 5 and 6, respectively.

Once both matrices BD and TW were generated, two matrices were stacked together, as shown in Figure 7. In Figure 7, the numbers in red present matrix BD, and the numbers in black present matrix WT. Subsequently, the only thing that needs to be conducted is to find a pixel pair (P1′, P2′) that is very similar to the original pixel pair and its coordinate maps to the values derived by Equations (8) and (9) in matrices BD and TW, respectively. To give a clear explanation, we will show an example of how the watermark embedding runs. Let the 24-bit watermark msg embedded in the block be {01110001, 00101011, 10000111}. Assume that we embedded the 3-bit watermark $\text{msg}_2 = \{\text{AC}_2, \text{RI1}_2, \text{RI2}_2\} = \{1, 0, 0\}$ in the second pair of the two pixels P3 = 229 and P4 = 225. The coordinate (x, y) is computed as (P3 mod 16, P4 mod 16) = (5, 1). From Figure 5, we find the coordinate (4, 1) in the clockwise direction from (5, 1) such that $\text{BD}_{4,\,1} = \text{AC}_2 = 1$ and $\text{WT}_{4,\,1} = 2 \times \text{RI1}_2 + \text{RI2}_2 = 0$. As a result, the embedded pair of pixels (P3′, P4′) is = (228, 225).



9	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}
8	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}
7	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}
6	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}
5	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}
4	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}
3	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}
2	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}
1	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}	0^{1}	1^{0}	2^{1}	3^{0}
0	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}	0^{0}	1^{1}	2^{0}	3^{1}
	0	1	2	3	4	5	6	7	8	9	10	11

Figure 7. Watermark embedding example.

3.2. Authentication and Recovery

The proposed fragile watermarking scheme provides authentication with tamper localization and self-recovery in the sense that if the image is deemed inauthentic, the regions of the image that have been tampered with can be localized and even reconstructed. The process of authentication and recovery is presented in Figure 8. We first calculate the authentication code $\overline{\text{AC}_{\text{org}}}$ for each block by using Equation (6) with the random seed γ from the preprocessing image, as described in Section 3.1. To extract a 3-bit watermark msg_i from a pair of pixels, let us read the next two pixels, P1′ and P2′, of a block as a pair in raster order. We compute the coordinate (x, y) as (P1′ mod 16, P2′ mod 16) and then obtain $\text{AC}_\text{i} = \text{BD}_{x,y}$, $\text{RI1}_\text{i} = \text{WT}_{\text{x, y}}/2$, and $RI2_i = WT_{x,y} \bmod 2$. For each block, the embedded msg1, msg2, . . . , msg8 are extracted to form the 24-bit watermark msg = $\{\overline{AC_{ext}}, \overline{RI1_{exr}}, \overline{RI2_{ext}}\}$.

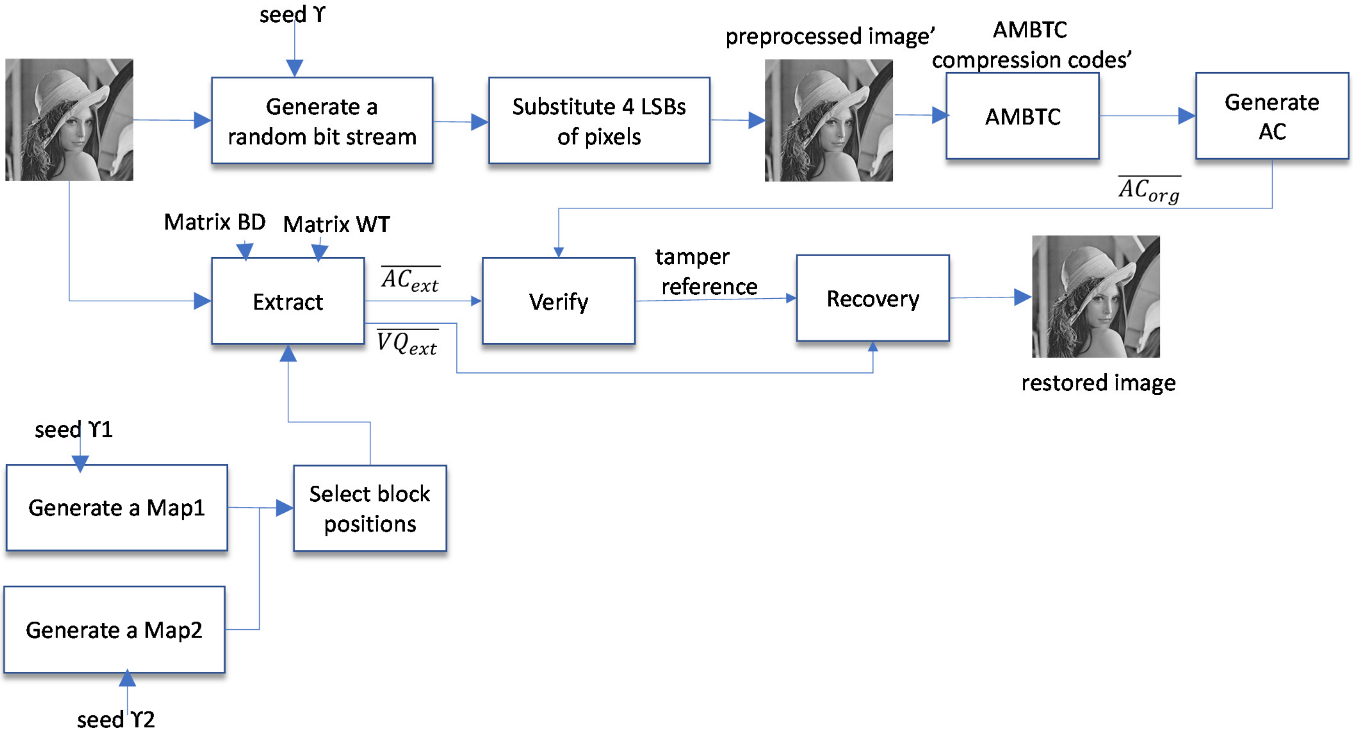


Figure 8. Authentication and recovery process.

The random seeds γ1 and γ2 are used to generate two non-repeating sequences: Map1 and Map2, respectively. For each block, we compare $\overline{AC_{org}}$ with $\overline{AC_{ext}}$ of its mapping block selected from Map1; if they are not equal, we mark it as invalid. To refine the tamper detection result, we took into account the block's surroundings. After the first round of tamper detection, we further marked the central block as invalid in these cases, as shown in Figure 9.

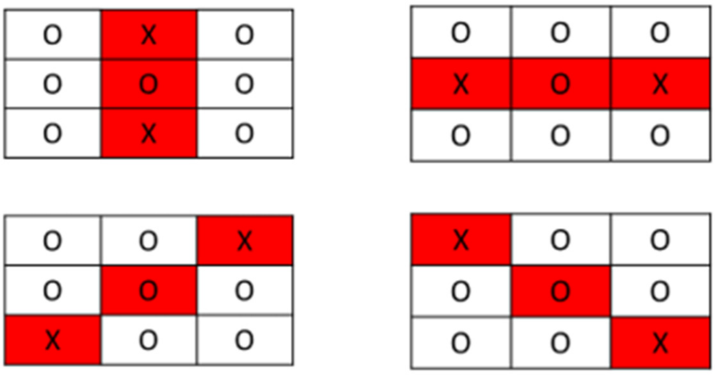


Figure 9. The refined tamper detection process.

In Figure 9, "x" indicates the current block is determined as "invalid." "o" indicates the current block is determined as "valid." A 3 × 3 mask shown in Figure 9 is used to judge whether the central block should be corrected as "invalid" by referring to its neighboring blocks. After tamper localization, we had to reconstruct the blocks that were marked as invalid. If the block was marked as invalid, we used $\overline{RI1_{exr}}$ of its mapping valid block selected from Map1 as the index of the VQ codebook to lookup the codeword to reconstruct it. If the mapping block selected from Map1 was invalid, we used $\overline{RI2_{ext}}$ of the mapping block selected from Map2 as the index of the VQ codebook to lookup the codeword to reconstruct the invalid block.

4. Experimental Results

A series of simulations were conducted to measure the performance of the proposed scheme in tamper localization and recovery. Note that we focus on a cutting attack and copy-paste attack. Four grayscale images, Lena, Elaine, Baboon, and Airplane, of 512 × 512 pixels, are shown in Figure 10 and were used to simulate the attacks. We also present a performance comparison with Lin et al.'s AMBTC-based scheme [23] in this section because Lin et al. designed a fragile watermarking mechanism based on AMBTC's features for the same spatial domain as ours. To further test the performance of our proposed scheme on tamper detection and recovery, 200 different test images were randomly selected from BOWS [26].

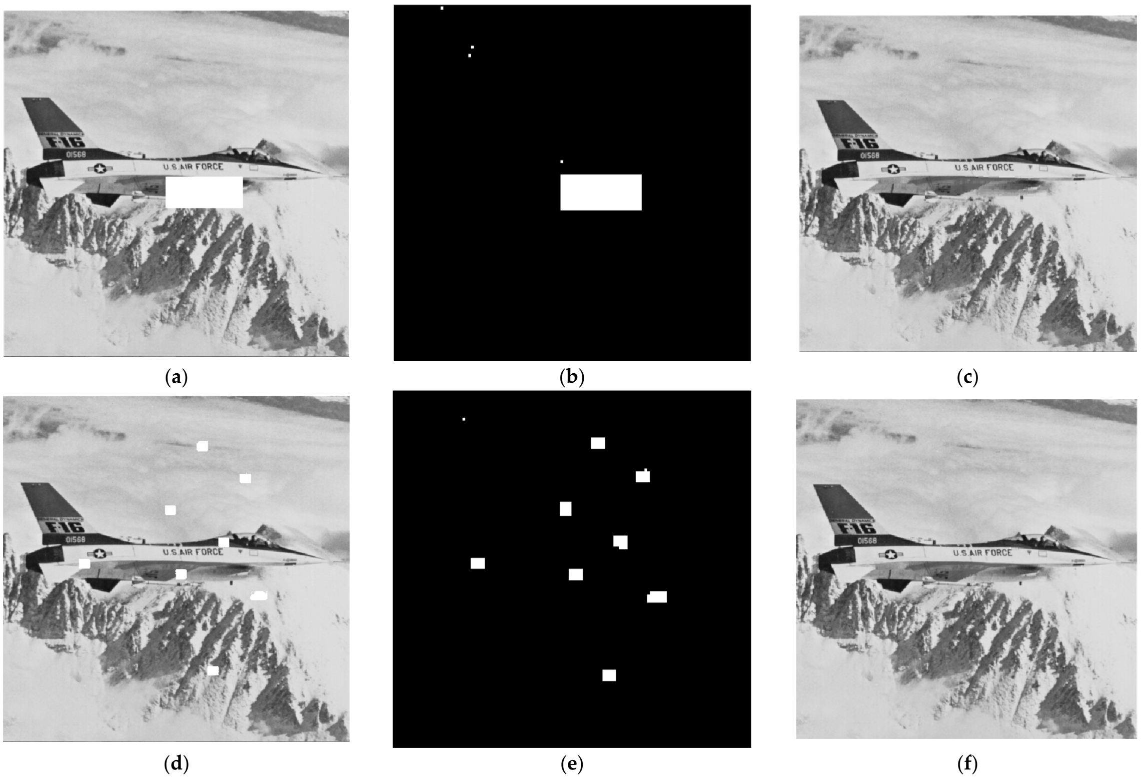
(**a**) Lena (**b**) Elaine

(**c**) Baboon (**d**) Airplane

Figure 10. Four original grey-scale images: (**a**) Lena, (**b**) Elaine, (**c**) Baboon, and (**d**) Airplane.

4.1. Attack Simulations

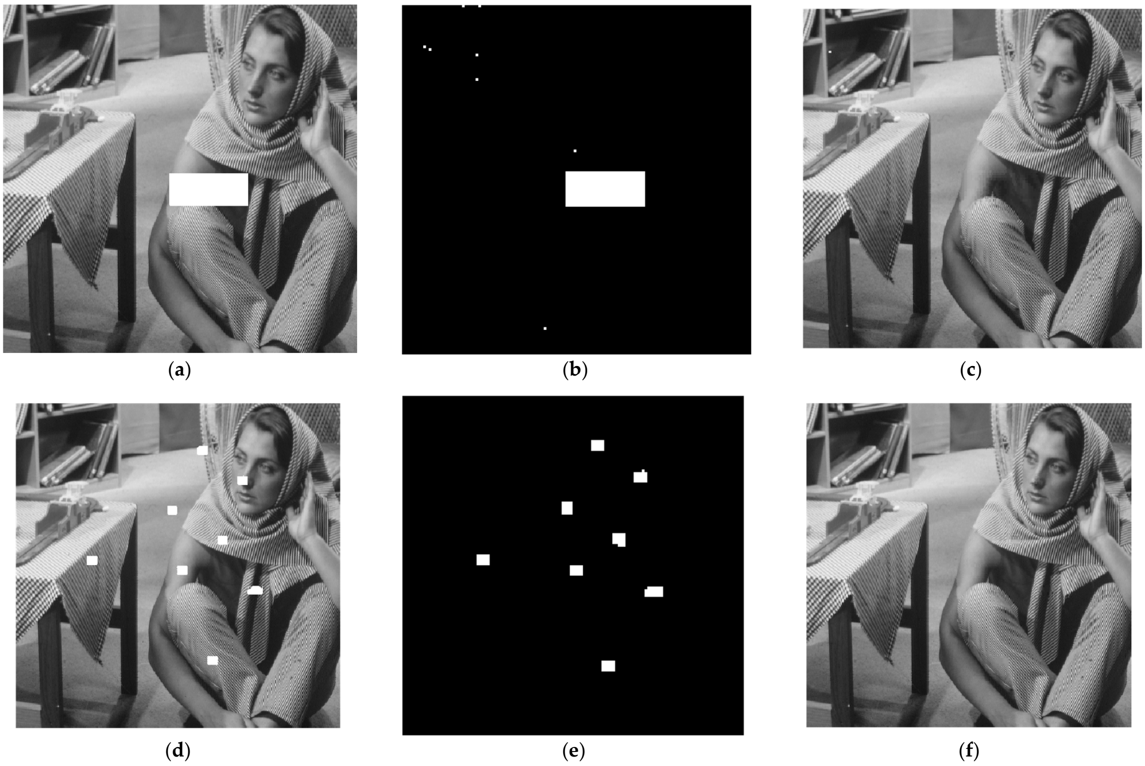
Two kinds of cutting attacks were simulated to test the performance of tamper localization and recovery. Figures 11 and 12 give the visual performance under the cutting attacks for Airplane and Barbara, respectively. The tampering rates α are set to be 2.04% and 0.74% for cutting attack 1 and cutting attack 2, respectively. We can detect general tampering; however, we can also detect some false detected blocks and also have some blocks of false acceptance. This is intuitively clear because the attacked blocks whose mapping block falls into the collaged region are not detected. We note that the tampering detection result has a large influence on recovery performance.



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

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

Figure 11. The visual performance for Airplane: (**a**) Cutting attack 1, (**b**) Detection of (**a**), (**c**) Recovery of (**a**), (**d**) Cutting attack 2, (**e**) Detection of (**d**), and (**f**) Recovery of (**d**).



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

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

Figure 12. The visual performance for Barbara: (**a**) Cutting attack 1, (**b**) Detection of (**a**), (**c**) Recovery of (**a**), (**d**) Cutting attack 2, (**e**) Detection of (**d**), and (**f**) Recovery of (**d**).

We further simulated two kinds of copy-paste attacks to measure the detection and recovery performance. The copy-paste attack copies the image blocks from one authenticated image and then inserts them into arbitrary positions in the watermarked image. Figures 13 and 14 show the visual performance under copy-paste attacks for Airplane and Barbara, respectively. The tampering rates α are set to be 4.53% and 1.84% for copy-paste attack 1 and copy-paste attack 2, respectively. The simulation results show that the regions tampered with by the copy-paste attack can be sufficiently detected and recovered.

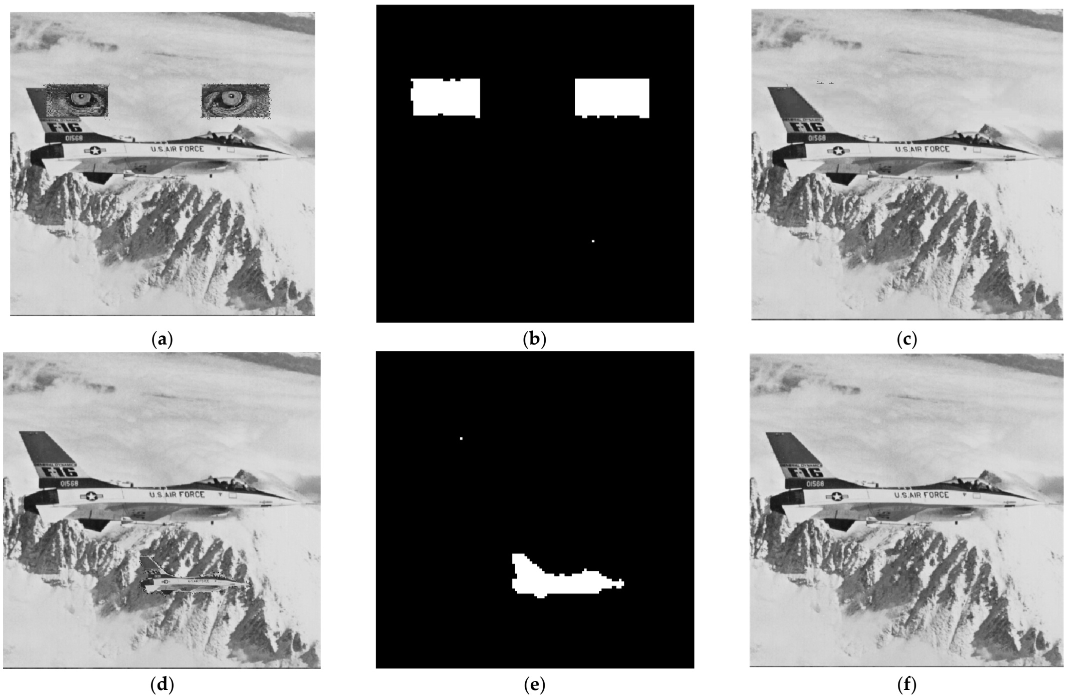


Figure 13. The visual performance for Airplane: (**a**) Copy-paste attack 1, (**b**) Detection of (**a**), (**c**) Recovery of (**a**), (**d**) Copy-paste attack 2, (**e**) Detection of (**d**), and (**f**) Recovery of (**d**).

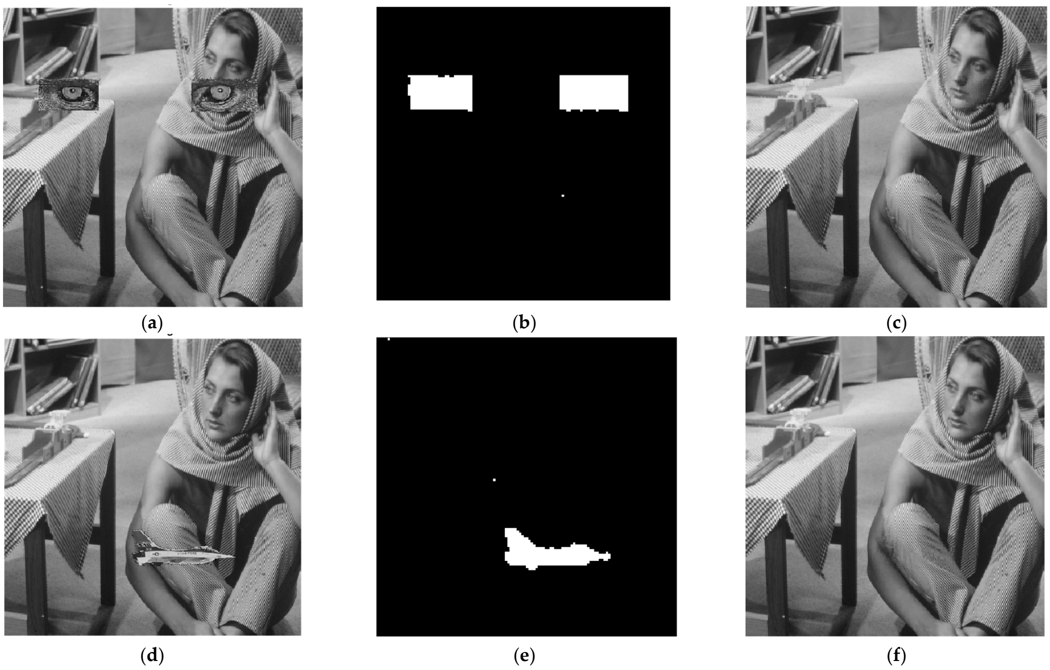


Figure 14. The visual performance for Barbara: (**a**) Copy-paste attack 1, (**b**) Detection of (**a**), (**c**) Recovery of (**a**), (**d**) Copy-paste attack 2, (**e**) Detection of (**d**), and (**f**) Recovery of (**d**).

4.2. Tamper Localization and Recovery Analysis

To measure the performance of tamper localization, we used true positive (TP), true negative (TN), false positive (FP), false negative (FN), true positive rate (TPR), and false positive rate (FPR) as quantitative measures. We also used a peak signal-to-noise ratio (PSNR) to quantify the reconstruction quality of the images. PSNR is calculated by summing up the squared differences between the evaluated image and the original image:

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\frac{1}{M\times N}\sum_{i=0}^{M}\sum_{j=0}^{N}(W(i,j)-O(i,j))^2} \tag{11}$$

where $W(i,j)$ and $O(i,j)$ denote the pixel values at location (i,j) and $M \times N$ is the image size.

Tables 1 and 2 give the tamper localization and recovery analysis against cutting attack 1 and cutting attack 2, respectively. Tables 3 and 4 show the tamper localization and recovery analysis against copy-paste attack 1 and copy-paste attack 2, respectively. Note that a high TPR essentially means a high probability that an actual positive will test positive, whereas FPR is the proportion of negative cases incorrectly detected as positive cases. Our proposed scheme achieves a high TPR and an acceptable FPR in the sense that most tampered regions can be identified, although a few blocks are incorrectly detected. Although most tampered regions can be correctly detected, it is clearly desirable to recover as much of the visual content of the original image as possible. It is inevitable that this will fail to recover the tampered block when the block and its mapping block are both tampered with by attacks. In Tables 1–4, our scheme introduces much less visible distortion into the image and, thus, has a high visual quality for the watermarked image and recovered image.

Table 1. Tamper localization and recovery analysis against cutting attack 1.

Images	α	TP	TN	FP	FN	TPR	FPR	Watermarked	Recovered
Lena	2.04%	6032	256048	0	64	0.9895	0	48.29 dB	43.70 dB
Elaine	2.04%	6032	256096	0	16	0.9973	0	48.33 dB	43.52 dB
Baboon	2.04%	6032	256096	0	16	0.9973	0	48.34 dB	43.67 dB
Airplane	2.04%	6032	256048	0	64	0.9895	0	48.21 dB	42.86 dB
Barbara	2.04%	6032	255984	0	128	0.9792	0	48.32 dB	41.54 dB

Table 2. Tamper localization and recovery analysis against cutting attack 2.

Images	α	TP	TN	FP	FN	TPR	FPR	Watermarked	Recovered
Lena	0.74%	2720	259376	16	32	0.9883	0.000062	48.29 dB	45.59 dB
Elaine	0.74%	2720	259392	16	16	0.9941	0.000062	48.33 dB	44.87 dB
Baboon	0.74%	2720	259360	16	48	0.9826	0.000062	48.34 dB	43.19 dB
Airplane	0.74%	2720	259376	16	32	0.9883	0.000062	48.21 dB	46.05 dB
Barbara	0.74%	2720	259392	16	16	0.9941	0.000062	48.32 dB	42.08 dB

Table 3. Tamper localization and recovery analysis against copy-paste attack 1.

Images	α	TP	TN	FP	FN	TPR	FPR	Watermarked	Recovered
Lena	4.53%	13824	248128	48	144	0.9896	0.000193	48.29 dB	41.43 dB
Elaine	4.53%	13776	248144	96	128	0.9907	0.000387	48.33 dB	39.59 dB
Baboon	4.53%	13824	248144	48	128	0.9908	0.000019	48.34 dB	34.52 dB
Airplane	4.53%	13744	248176	128	96	0.9930	0.000515	48.21 dB	42.54 dB
Barbara	4.53%	13776	248176	96	96	0.9930	0.000387	48.32 dB	39.29 dB

Table 4. Tamper localization and recovery analysis against copy-paste attack 2.

Images	α	TP	TN	FP	FN	TPR	FPR	Watermarked	Recovered
Lena	1.84%	6096	255856	0	192	0.9694	0	48.29 dB	45.31 dB
Elaine	1.84%	6096	255872	0	176	0.9719	0	48.33 dB	43.92 dB
Baboon	1.84%	6080	255856	16	192	0.9693	0.000063	48.34 dB	40.93 dB
Airplane	1.84%	6096	255888	0	160	0.9744	0	48.21 dB	41.49 dB
Barbara	1.84%	6096	255872	0	176	0.9719	0	48.32 dB	38.06 dB

It is noted that in Tables 1–4, data listed in the "watermarked" column show the image quality after concealing both the authentication code and recovery information. Therefore, the data listed in the "watermarked" column are the same. This is because no attacks occurred at this stage; the authentication codes are derived from the original image, and the recovery information is also derived from the original image. As for the data listed in the "Recovered" column, since attack types are different, the PSNRs listed in the "Recovered" column are similar but not the same.

To further evaluate our performance for tamper detection and recovery, 50 different images were randomly selected from the representative database BOWS [26] and used in two sets of experiments. In Figures 15 and 16, the PSNRs of the watermarked images and recovered images against cutting attack 1 and copy-past attack 2, respectively, are presented. Since each figure presents 50 different images randomly selected from the BOWS database BOWS, there are 100 different images that were examined. From these two figures, our double matrix encoding strategy worked well, such that the PSNRs of 100 different watermarked images are very consistent and are close to 50 dB. For copy-past attack 2, the corresponding image quality of the restored images is almost higher than 40 dB.

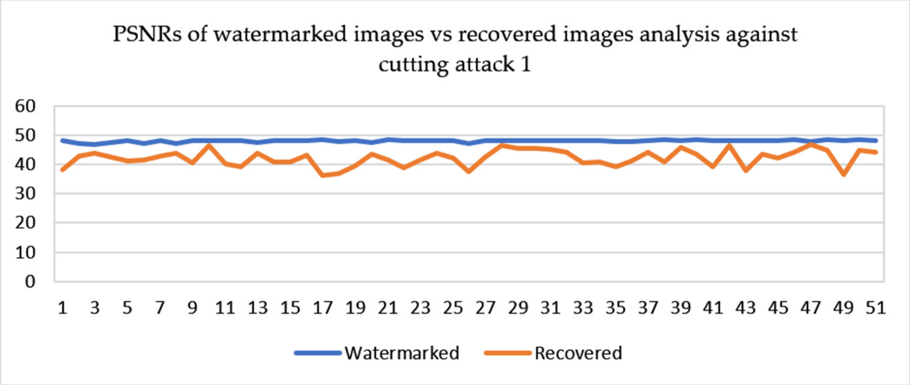


Figure 15. PSNRs of watermarked images vs. recovered image analysis against cutting attack 1.

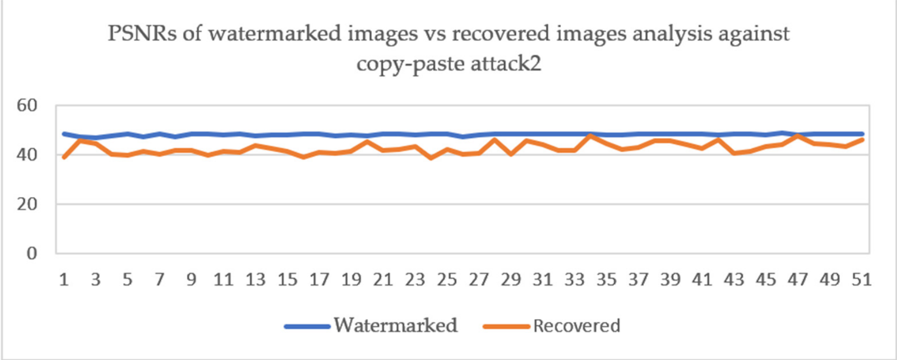


Figure 16. PSNRs of watermarked images vs. recovered image analysis against copy-past attack 2.

To demonstrate the performance of our scheme on tamper detection and recovery against four types of attacks, 200 different images were randomly selected from the BOWS database [26] and then divided into four sets and tested through four types of attacks, as shown in Figures 9 and 12, respectively. The related comparisons are listed in Table 5. From Table 5, the average TPR is around 96.09%, but the average FPR is about 0.03%. It is confirmed that the hidden authentication codes derived from the AMBTC compression codes successfully enhanced our tamper detection performance. For attack 2a, which is copy-paste attack 1, due to the relatively larger attack area, the image quality of the restored image is less than others.

Table 5. Performance comparison of tamper detection and image quality against four types of attacks.

Attacks	α	TP	TN	FP	FN	TPR	FPR	Watermarked	Recovered
Attack 1a	2.04%	5984	255998	48	114	98.16%	0.02%	48.02	42.13
Attack 1b	0.74%	2611	259244	125	164	94.16%	0.05%	48.07	43.58
Attack 2a	4.53%	13792	248138	80	134	99.04%	0.03%	48.02	38.08
Attack 2b	1.84%	6072	255862	24	186	97.03%	0.01%	48.02	42.55

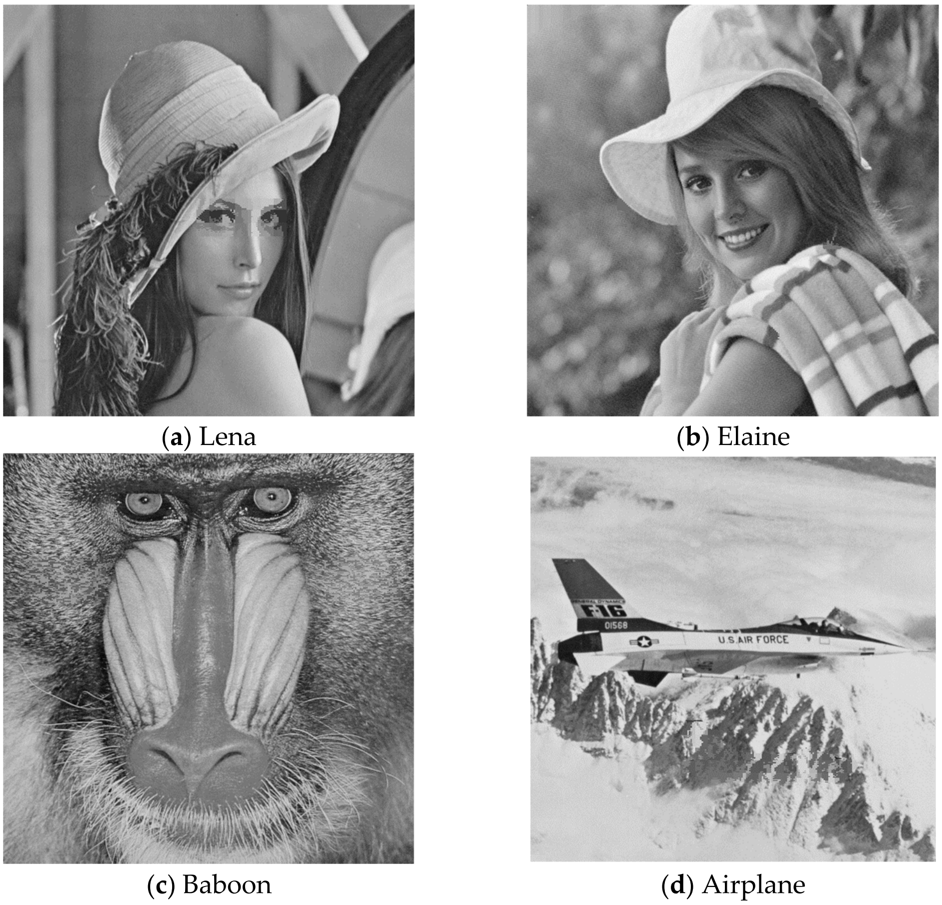
Table 6 shows that the image quality of the restored images is related to the tamper ratio. However, even in the worst case, the restored images remain at 31.26 dB on average. Considering that Lin et al.'s AMBTC-based fragile watermarking [23] is the latest work that is based on AMBTC, the comparisons of the performance on tamper detection and recovery between Lin et al.'s AMBTC-based fragile scheme [23] and ours are demonstrated in Table 7. The recovered images from our scheme are also depicted in Figure 17. Comparing them with the originals shown in Figure 8, the recovered images are the same from the perspective of human visualization.

Table 6. Performance comparison of tamper detection and image quality against two types of attacks with different attack ratios.

Attacks	α	TP	TN	FP	FN	TPR	FPR	Watermarked	Recovered
Attack 1a	2.04%	5984	255998	48	114	98.16%	0.02%	48.02	42.13
Attack 1a	4.36%	11932	248304	756	1152	91.54%	0.30%	48.06	36.49
Attack 2a	4.53%	13792	248138	80	134	99.04%	0.03%	48.02	38.08
Attack 2a	9.61%	28205	232647	1139	153	99.46%	0.49%	48.08	31.26

Table 7. Performance comparison with Lin et al.'s AMBTC-based scheme [23].

Schemes	Images	α	TP	TN	FP	FN	TPR	FPR	Watermarked	Recovered
Proposed	Lena	2.04%	6032	256064	0	48	0.9921	0	48.29 dB	44.12 dB
	Elaine	0.74%	2736	259392	0	16	0.9941	0	48.32 dB	45.35 dB
	Baboon	4.53%	13824	248128	48	144	0.9896	0.00019	48.33 dB	34.95 dB
	Airplane	1.84%	6096	255888	0	160	0.9744	0	48.20 dB	42.69 dB
Lin et al. [23]	Lena	1.93%	4950	256882	110	202	0.9608	0.0004	46.8 dB	37.9 dB
	Elaine	0.6%	1447	260183	121	393	0.7864	0.0004	46.8 dB	41.7 dB
	Baboon	3.44%	8740	251707	287	1410	0.8611	0.0011	46.8 dB	31.8 dB
	Airplane	1.41%	3229	257870	459	586	0.8464	0.0018	46.8 dB	35.5 dB


(**a**) Lena

(**b**) Elaine

(**c**) Baboon

(**d**) Airplane

Figure 17. Four recovered images: (**a**) PSNR = 44.12 dB, (**b**) PSNR = 45.35 dB, (**c**) PSNR = 34.95 dB, and (**d**) PSNR = 42.69 dB.

Based on the above data, listed in Table 7, we can see that Lin et al.'s scheme [23] has a lower TPR and higher FPR than our scheme, which indicates that their scheme has a higher probability of incorrectly detecting tampered regions. A possible drawback of Lin's scheme is the fact that some blocks are deemed authentic, whereas it is actually tampered with by attacks, or some valid blocks are detected as invalid. Lin's scheme also shows a larger distortion in the watermarked image and recovered image, introducing easily detectable artifacts. Our scheme can effectively detect the tampered blocks with a high TPR and a low FPR under the cutting attack and the copy-paste attack.

To further demonstrate the image quality of the watermarked image and restored image under the maximum tolerable tampering rate, Table 8 shows comparisons of theoretical values for the proposed scheme and eleven existing schemes [3,7,9,12–15,20–23] with four representative test images shown in Figure 10. In Table 8, the column "PSNR of recovered image" indicates the average image quality of the recovered results from the tampered images. The column "Condition of recovery" demonstrates the conditions for successful recovery in all the schemes, i.e., maximum tolerable tampering rates. Overall, the highest image quality of the watermarked images offered by our scheme cannot compete with that offered by Yang et al.'s scheme [9], but our restored image quality and tolerable tampering rate are relatively higher than Yang et al.'s scheme. The maximum tolerable tampering rates of [8,15] are less than 30%, which means that the restored image failed to be restored when the tampering rate was up to 30%. As for the remaining schemes, our proposed scheme is included, and the maximal tampering rate can be close to 50% or even close to 59%, as was Zhang et al.'s scheme [3]. Therefore, based on the experimental data listed in Table 7, our proposed scheme makes progress at the tradeoff between the image quality of a watermarked image, restored image, and maximal tampering rate by combining VQ and AMBTC.

Table 8. Performance comparison of proposed scheme versus eleven schemes.

Schemes	PSNR of Watermarked Images	PSNR of Recovered Images	Condition of Recovery	Attack Types
Zang et al. [3]	37.9 dB	[26, 29] dB	$\propto < 59\%$	copy-paste
Zang et al. [7]	37.9 dB	40.7 dB	$\propto < 24\%$	copy-paste
Yang et al. [9]	51.3 dB	[24, 36] dB	$\propto <50\%$	copy-paste
Qin et al. [12]	[37.92, 51.14] dB	[40.74, 51.12] dB	$\propto <20\%$	cutting/copy-paste
Cat et al. [13]	[44.11, 44.17] dB	[36.90, 46.34]	$\propto <20\%$	cutting
Qin et al. [14]	44.15 dB	[30.36, 36.21] dB	$\propto <60\%$	copy-paste
Lin et al. [15]	37.9 dB	$+\infty$	$\propto < 26\%$	copy-paste
Roy et al. [20]	[37.92, 54.13] dB	[28.63, 46.98] dB	$\propto < 50\%$	rotation/ salt & pepper [1]
Hong et al. [21]	[43.19, 47.04] dB	[21.31, 41.77] dB	$\propto < 15\%$	copy-paste
Chang et al. [22]	49.76 dB	34.65 dB	$\propto < 50\%$	cutting, copy-paste
Lin et al. [23]	46.8 dB	[32, 42] dB	$\propto < 50\%$	copy-paste
Proposed scheme	[48.21, 48.34] dB	[34.95, 44.12] dB	$\propto < 50\%$	cutting, copy-paste

Note [1]. Roy et al.’s watermarking is the robust watermarking; they tested five attacks, such as rotation, salt and pepper, median filtering, scaling, and JPEG compression attacks, to prove their robustness.

Considering that there are multiple fragile watermarking schemes based on BTC/AMBTC or VQ, which are combined and adopted in our scheme, we present a comparison between these existing schemes and ours in Table 9. It can be noted that the PSNRs of the watermarked image listed in Table 9 are directly cited from the corresponding works with four representative test images shown in Figure 10.

Table 9. Comparison between proposed and other BTC/AMBTC/VQ-based fragile watermarking schemes.

Schemes	Image Type	PSNR of Watermarked Image (dB)	AC Code	AC Embedding	Recovery
Lee et al. [5]	Spatial images	39.6	SVD codes, random stream	LSB substitution/pixels	Yes
Lin et al. [8]	AMBTC-compressed images	34.11	Random stream, preprocessed bitmap	Substitution/quantization level(s)	No
Li et al. [11]	BTC-compressed images	41.62	Random stream	Matrix encoding/quantization levels	No
Qin et al. [12]	Spatial images	44.15	A hash value of VQ indices and original image	LSB substitution/pixels	Yes
Lin et al. [15]	AMBTC-compressed images	37.9	Random stream	Hybrid embedding/quantization levels and bitmap	No
Hong et al. [17]	AMBTC-compressed images	30.41 [1]	A hash value of bitmap and MSBs of two quantization levels	LSB substitution/quantization levels	No
Chang et al. [18]	AMBTC-compressed images	32.60	Bitmap and random stream	Turtle shell data hiding/ quantization levels	No
Su et al. [19]	AMBTC-compressed images	31.99	Random stream	Matrix encoding/ bitmap	Yes
Hong et al. [21]	AMBTC-compressed images	44.51	A hash value of flipped bitmaps	APPM/quantization levels	Yes
Chang et al. [22]	Spatial images	49.76	Compression code of bitmap	Turtle shell data hiding/pixel pair	Yes
Lin et al. [23]	Spatial images	46.8	RI derived from AMBTC compression codes	Matrix encoding/pixel pair	Yes
Proposed scheme	Spatial images	48.21	Hash value of AMBTC compression codes	Double matrix encoding/pixel pair	Yes

Note:[1] It is the PSNR offered by Hong et al.’s MSBP.

Based on the above comparisons, we can see that most AMBTC/BTC-based fragile schemes are designed for AMBTC/BTC compressed images, and only Qin et al. [11], Chang et al. [21], Lin et al. [22], and our scheme were designed for spatial images and the features derived from AMBTC or BTC compression codes are utilized as an authentication code. Although Chang et al. [21] had the highest image quality for a watermarked image at 49.76 dB, our scheme has the second-highest PSNR at 48.21 dB. However, the average PSNR of the restored image with Chang et al.'s scheme [21] is only 34.65 dB, which is relatively lower than ours. Combining Tables 8 and 9, only Roy et al.'s scheme [19] is a robust watermarking scheme, and they tested five different attacks to prove robustness. Among the remaining fragile watermarking schemes, our scheme not only tests against the cutting attack but also a copy-paste attack which is also called a collage. With AMBTC compression codes serving as the authentication code in our scheme, the number of authentication bits is larger than other methods, and accordingly, our average TPR rate is more than 97% with 200 different test images randomly selected from BOWS while maintaining a 0.12% FPR.

5. Conclusions

In this paper, we proposed an authentication technique with the capabilities of tamper localization and self-recovery by combining VQ and AMBTC. To improve tamper detection capability AMBTC compressed codes were applied as the authentication code. Additionally, with VQ indices derived from the original image, VQ serves as the recovery information for the tampered regions. Since the number of hidden authentication bits is relatively larger than other existing schemes, to reduce the distortion caused during the concealment of the authentication code and recovery information, a double matrix encoding method was proposed to minimize the introduction of obvious artifacts. Experimental results based on certain representative test images, or 200 test images randomly selected from BOWS confirmed that the proposed scheme could effectively resist a cutting attack and a copy-paste attack while retaining high accuracy for tamper localization. The average TPR and average FTP rate were around 97% and 0.12%, respectively, indicating that the general tamper detection performance is confirmed while maintaining the image quality of the watermarked image and restored image at up to 48 dB and 39.28 dB, respectively. The experimental results also show that there is room to improve the image quality of the restored images. Therefore, in future work, we will seek either other VQ variants or other techniques to further improve recovery performance.

Author Contributions: Conceptualization and methodology, C.-C.L. and P.-F.S.; software, P.-F.S. and B.Z.; validation, Y.-F.C., T.-L.L. and C.-C.L.; writing—original draft preparation, P.-F.S., Y.-F.C. and B.Z.; writing—review and editing, C.-C.L. and T.-L.L. Funding Acquisition, C.-C.L. and Y.-F.C. All authors have read and agreed to the published version of the manuscript.

Funding: National Science and Technology Council Taiwan, MOST 111-2221-E-025 -007; MOST 110-2410-H-167 -004.

Data Availability Statement: Not applicable.

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

References

1. Fridrich, J.; Goljan, M.; Memon, N. Cryptanalysis of the Yeung-Mintzer fragile watermarking technique. *J. Electron. Imaging* **2002**, *11*, 262–274.
2. Chang, C.C.; Fan, Y.H.; Tai, W.L. Four-scanning attack on hierarchical digital watermarking method for image tamper detection and recovery. *Pattern Recognit.* **2008**, *41*, 654–661. [CrossRef]
3. Zhang, X.; Wang, S.; Feng, G. Fragile watermarking scheme with extensive content restoration capability. In *International Workshop on Digital Watermarking*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 268–278.
4. Zhang, X.; Wang, S.; Qian, Z.; Feng, G. Reference Sharing Mechanism for Watermark Self-Embedding. *IEEE Trans. Image Process.* **2011**, *20*, 485–495. [CrossRef]
5. Lee, C.F.; Chen, K.N.; Chang, C.C.; Tsai, M.C. A Hierarchical Fragile Watermarking with VQ Index Recovery. *J. Multimed.* **2011**, *6*, 277–283. [CrossRef]
6. He, H.; Chen, F.; Tai, H.; Kalker, T.; Zhang, J. Performance analysis of a block-neighborhood-based self-recovery fragile watermarking scheme. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 185–196. [CrossRef]
7. Zhang, J.; Zhang, Q.; Lv, H. A novel image tamper localization and recovery algorithm based on watermarking technology. *Optik* **2013**, *124*, 6367–6371. [CrossRef]

8. Lin, C.C.; Huang, Y.H.; Tai, W.L. A high-quality image authentication scheme for AMBTC-compressed images. *KSII Trans. Internet Inf. Syst.* **2014**, *8*, 4588–4603.
9. Yang, S.; Qin, C.; Qian, Z.; Xu, B. Tampering detection and content recovery for digital images using halftone mechanism. In Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan, 27–29 August 2014; pp. 130–133.
10. Sarreshtedari, S.; Akhaee, M.A. A source-channel coding approach to digital image protection and self-recovery. *IEEE Trans. Image Process.* **2015**, *24*, 2266–2277. [CrossRef]
11. Li, W.; Lin, C.-C.; Pan, J.-S. Novel image authentication scheme with fine image quality for BTC-based compressed images. *Multimed. Tools Appl.* **2015**, *75*, 4771–4793. [CrossRef]
12. Qin, C.; Wang, H.; Zhang, X.; Sun, X. Self-embedding fragile watermarking based on reference-data interleaving and adaptive selection of embedding mode. *Inf. Sci.* **2016**, *373*, 233–250. [CrossRef]
13. Cao, F.; An, B.; Wang, J.; Ye, D.; Wang, H. Hierarchical recovery for tampered images based on watermark self-embedding. *Displays* **2017**, *46*, 52–60. [CrossRef]
14. 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]
15. Lin, C.C.; Huang, Y.H.; Tai, W.L. A novel hybrid image authentication scheme based on absolute moment block truncation al. coding. *Multimed. Tools Appl.* **2017**, *76*, 463–488. [CrossRef]
16. Tai, W.L.; Liao, Z.J. Image self-recovery with watermark self-embedding. *Signal Process. Image Commun.* **2018**, *65*, 11–25. [CrossRef]
17. Hong, W.; Zhou, X.; Lou, D.C.; Huang, X.; Peng, C. Detectability improved tamper detection scheme for absolute moment block truncation coding compressed images. *Symmetry* **2018**, *10*, 318. [CrossRef]
18. 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]
19. Su, G.; Chang, C.C.; Lin, C.C. High-precision authentication scheme based on matrix encoding for AMBTC-compressed images. *Symmetry* **2019**, *11*, 996. [CrossRef]
20. Roy, S.S.; Basu, A.; Chattopadhyay, A. On the implementation of a copyright protection scheme using digital image watermarking. *Multimed. Tools Appl.* **2020**, *79*, 13125–13138.
21. Hong, W.; Wu, J.; Lou, D.C.; Zhou, X.; Chen, J. An AMBTC authentication scheme with recoverability using matrix encoding and side match. *IEEE Access* **2021**, *9*, 133746–133761. [CrossRef]
22. Chang, C.-C.; Lin, C.-C.; Su, G.-D. An effective image self-recovery based fragile watermarking using self-adaptive weight-based compressed AMBTC. *Multimed. Tools Appl.* **2020**, *79*, 24795–24824. [CrossRef]
23. Lin, C.C.; He, S.L.; Chang, C.C. Pixel pair-wise fragile image watermarking based on HC-based absolute moment block truncation coding. *Electronics* **2021**, *10*, 690. [CrossRef]
24. Lema, M.; Mitchell, O. Absolute moment block truncation coding and its application to color images. *IEEE Trans. Commun.* **1984**, *32*, 1148–1157. [CrossRef]
25. Gray, R.M. Vector quantization. *IEEE Trans. Acoust. Speech Signal Process.* **1984**, *1*, 4–29. [CrossRef]
26. Bas, P.; Furon, T. Image Database of BOWS-2. Available online: http://bows2.ec-lille.fr/ (accessed on 20 June 2017).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.