

Article

# Self-Embedding Authentication Watermarking with Effective Tampered Location Detection and High-Quality Image Recovery

Chin-Feng Lee <sup>1</sup>, Jau-Ji Shen <sup>2</sup>, Zhao-Ru Chen <sup>2</sup> and Somya Agrawal <sup>1,\*</sup>

<sup>1</sup> Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan; lcf@cyut.edu.tw

<sup>2</sup> Department of Management Information Systems, National Chung Hsing University, Taichung 40227, Taiwan; jjshen@nchu.edu.tw (J.-J.S.); g105029023@mail.nchu.edu.tw (Z.-R.C.)

\* Correspondence: asomya@gm.cyut.edu.tw

Received: 25 February 2019; Accepted: 9 May 2019; Published: 16 May 2019



**Abstract:** Recently, sensor networks have emerged as a high-impact research area, and a number of high profile applications have been proposed. Although significant progress has already been made on securing basic network protocols, additional research is needed to produce techniques and methods for protecting canonical tasks in wireless sensor networks. In this paper, we propose an effective self-embedding authentication watermarking method for tampered location detection and image recovery. The proposed detection method is classified into block-wise and pixel-wise. In block-wise detection, if the size of the block is small, the false positive rate (FPR) will be low. In pixel-wise detection, when the tampered pixels are detected, only the corresponding pixel area is marked. Therefore, the FPR will be lower than that of the block-wise detection. The experimental results demonstrate that the proposed method was effective, and accurate tamper detection and high-quality recovery can be realized even in highly tampered images.

**Keywords:** wireless sensor networks; image stenography; image authentication scheme; tamper detection; fragile watermarking method; self-recovery

---

## 1. Introduction

Recently, sensor networks have emerged as a high-impact research area and a number of high profile applications have been proposed. Sensors are usually distributed in a sensory field and are used for applications, such as smart home, environmental monitoring, battlefield surveillance, information collection, etc. [1]. Wireless sensor networks (WSN) are distributed embedded systems where each unit is equipped with a defined amount of computation, communication, storage, and sensing resources. Such sensor networks have the capacity to store information not only about one or more users but they also contain a great deal of information about their past and even future actions. Moreover, once the sensors have been equipped with actuators, both the sensors and the environment can be impacted in a number of ways. However, WSNs are highly prone to security attacks intrinsically due to their deployment, their hardware, and their resource constraints. They are often deployed in uncontrolled and sometimes even hostile settings. The wireless communication networks on a large scale can be easily observed and interfered with. It is possible to manipulate the sensor networks even without interfering with the electronic subsystem of the node and actuators which can pose strong safety and hazard concerns. In addition, they have constraints in terms of energy and, therefore, extensive on-line security checking is not feasible. Therefore, wireless sensor network (WSN) nodes are complex component systems with numerous weak points from a security point of view. The role of security in

WSN is highly important, and a number of security and privacy issues need to be addressed, such as how to ensure the integrity of sensor data, how to provide mechanisms for authentication and access control [2]. There is an urgent need to develop methods that ensure privacy of subjects and objects in the sensor networks. With watermarking techniques, the image or even video can be protected. We can detect the precise location of the tampered image or which frame of the video was damaged. Other methods, such as encryption, are unable to do so. The encryption method can only know that the file has been tampered rather than detect the damaged location.

Although significant progress has already been made in securing basic network protocols, additional research is needed to produce techniques and methods for protecting canonical tasks in wireless sensor networks, such as routing, broadcast, multicast, and data aggregation. WSN require new concepts, techniques, and methods with respect to security, privacy, digital rights management, and usage measurement [3]. The Internet has been a great facilitator of computer and communication security on a large scale. However, it has itself created opportunities for new types of attacks, such as denial of service (DoS) and intrusion detection. In sensor networks, watermarking and other intellectual property protection techniques can be used at a variety of levels. Software used in the network and the design of sensor nodes can be protected using functional techniques. Both static and functional watermarking techniques [4] can be applied to the data collected from the network depending on the types of sensors and actuators deployed (i.e., video, audio, measured data).

In particular, image authentication technology is classified into two main types: the digital signature-based method and the watermarking method. Digital signatures are always stored by third parties in digital signature-based methods. In this approach, the digital signature retrieved from an image is compared with the digital signature stored by the third party. Comparing the two signatures helps us to detect whether the image has been tampered with or not [5–8]. The watermarking method can be categorized into robust watermarking [9–15], semi-fragile watermarking [16–19] and fragile watermarking [20–34]. In the robust watermarking method, hidden watermarks can be retrieved from the watermarked images after they have undergone image processing, such as noise processing or image compression. It can be used to validate copyrights and intellectual property rights. Fragile watermarks are hidden watermarks in the image that can be easily destroyed by tampering, and, thus, can also be accurately detected. Currently, there are two types of fragile watermarking technologies. The first type can only detect tampered digital images and locate the tampered areas. On the other hand, the second type can not only detect and locate the tampered areas, but it can also recover damaged areas in the image. The second type of image authentication technology can detect and locate tampered areas as well as do a self-recovery of the tampered areas, which makes it extremely helpful for the protection of image integrity. The present study utilizes this image authentication technology and proposes an effective self-embedding watermarking method that can not only detect an image's tampered area accurately, but it can also improve image recovery capabilities of the tampered area effectively.

The remainder of this paper is organized as follows: In Section 2, we will introduce several methods that have been proposed in previous studies. Section 3 describes the proposed method that is classified as block-wise detection and pixel-wise detection. Experimental results and comparisons will be presented in Section 4 to demonstrate that the proposed method is more effective and better than other methods. Finally, Section 5 includes the conclusion of this paper.

## 2. Related Work

To understand the development and application of today's fragile watermarking technologies, this section provides a brief on fragile watermarking technologies proposed by various scholars in the spatial, compression, and frequency domains in the recent years. Two types of data are generated by a sensor network: raw sensor data and processed application data. The aim here is to watermark all data provided by wireless sensor networks. The first type of sensor data is the original data the sensor network measures and the second type, processed data, is the output of the network. The second

type of data is the information the user of the network expects from the network. The distinction of these two types of data [35] gives us a hint on where watermarking can take place: (i) during the process of sensing data (original data capturing); (ii) during the process of processing the original data. Since most image tampering detection methods are based on image blocks and ignore characteristics of the image blocks, it results in poor image quality after hiding the watermark. Therefore, in 2011 Lee et al. [16] proposed a semi-blind watermark scheme exploiting a self-reference image by using just noticeable distortion (JND) approach for digital image protection and authenticity. Hsu and Tu [23] in 2016 proposed adaptive embedding rules for the detection and recovery of image tampering that apply different hiding, detection, and recovery methods according to the block's level of smoothness. The method mentioned above is implemented in the image's spatial domain during the watermark generation process. If the recovery information is generated in the frequency domain, the quality of the recovered image will be much higher. In 2014, Lo and Hu [27] proposed a new reversible image authentication method which uses pseudo-random numbers to generate authentication data to serve as watermarks and uses a prediction-based histogram shifting scheme to embed the watermark into the original image. The complete watermark can be retrieved if the watermarked image has not been tampered with, and the watermarked image can be restored to the original image. In 2016, Singh and Singh [29] proposed an effective self-embedding watermarking method for image tamper and recovery capability positioning. This method uses Discrete Cosine Transformation (DCT) to generate authentication and recovery data from the image. During the embedding process, authentication data will be embedded in the block itself, and recovery bits will be embedded in mapping blocks, which increases the performance of image tamper detection. Because DCT is used to generate recovery data, the image quality will be better than the previous methods used in the past. Moreover, some studies have found that original images were unrecoverable if irreversible data hiding methods were used to embed watermarks. Therefore, reversible data hiding methods are used mostly to embed watermarks. The function of this method is that the original image can be recovered if the watermarked image has not been tampered with. However, it can only locate the tampered area and cannot recover the image.

In 2016, Yin et al. [30] proposed a method to improve Lo and Hu's approach [27]. It uses a Hilbert Curve to scan the entire image and allows neighbor pixel values to group together. Pixel value ordering (PVO) is then used to embed the watermark. This method enables better control of the changes in pixel value and watermark can be embedded according to the block's complexity as well.

In 2017, Qin et al. [31] proposed a fragile image watermarking method with pixel-wise recovery based on overlapping embedding strategy. First, the original image  $I_O$  (with a size of  $w \times h$ ) is divided into overlapping blocks  $B$  with a size of  $3 \times 3$ . Every block  $B_{m,n}$  ( $m = 1, 2, \dots, h/2$ ,  $n = 1, 2, \dots, w/2$ ) should consist of three pixels that overlap with neighboring blocks. Then the embedding rule is applied which requires that every center pixel of each block should contain the authentication data  $A$ , and either one or two bits of the recovery data needs to be embedded in the pixel. The complexity level of the block is calculated, and two thresholds levels are used to determine whether the block is complex, general or smooth. Two to four bits of authentication data have different authentication data  $A$  according to their degree of complexity. The authentication data  $A$  is then generated by using the Hash function of the pixel value, block number, as well as the image's name or ID. Its length is set according to the degree of complexity of the block. The generation of the recovery data  $R$  is then calculated by averaging the pixel value of  $M$  and its six most significant bits as  $M'$ , and further combined with the entire  $M'$  image as  $U$  with a length of  $6 \times (h \times w/4)$ . Thereafter,  $U$  is then scrambled and divided into  $k$  sub-sets by using the SK secret key. Each set contains four bits of recovery data  $R$  for each block. Authenticated and recovery data are then embedded in each block.

In 2018, Tai and Liao [33] proposed an efficient block-based fragile watermarking scheme for image tamper detection and self-recovery. In their method, the authentication and the recovery information were both generated by using the wavelet transform rather than the common block average method. The 4-bit authentication code was generated by using the low-frequency sub band of Haar wavelet transform of each block, and the 28-bit recovery code was produced by the low-frequency sub band

and two high-frequency sub bands. To break the independency of each block the method used block mapping technique called Arnold's cat map transform to scramble the image blocks. The authentication code and recovery code were then embedded from other blocks into each block.

Each of the five fragile watermarking techniques discussed above has their own advantages and disadvantages. The two methods of Lo and Hu [27] and Yin et al. [30] made use of reversible data hiding techniques to embed a watermark into the image so that the watermarked image can be restored to original image when no tampering occurs in the image. However, these two methods do not provide any recovery scheme to restore the tampered images, which means that they do not have the ability to recover the original image after the image has been tampered. The method proposed in this paper has designed a recovery scheme using mean value and block enlarger techniques to resolve this issue. In another method proposed by Singh and Singh [29], they used DCT (Discrete Cosine Transformation) to generate authentication and recovery information. The advantage of using DCT is that the recovered image quality will better as compared to other methods which just use the mean value to restore the original image. The methods proposed by Hsu and Tu [23] also have the advantage of better image quality of the recovered image as their methods use different procedures to recover information based on the level of smoothness of the block in the image. However, the experimental results of our proposed method have shown better performance in terms of image quality of the recovered image when compared to Singh and Singh's [29] and Hsu and Tu's [23] methods. The method of Qin et al. [31] can only detect the tamper when the tamper ratio is below 45%, and the proposed method not only enhances the recover image quality than the previous studies but also resolve the tamper detection issue of the Qin et al. [31] method. The method proposed by Tai and Liao have better recovery performance compared to other methods. However, when dealing with highly complex images their peak signal to noise ratio (PSNR) of recovered image is not quite good. Our proposed method overcomes this limitation.

### 3. Proposed Method

The image authentication technology proposed by our method can detect and locate tampered area as well as recover the tampered area. To detect tampered areas more accurately and improve their recovery effectively, our method proposes image authentication technology based on block-wise and pixel-wise detection methods. According to the experimental results, the error rate of the pixel-wise method is lower than the block-wise method. The quality of images recovered from using either pixel-wise or block-wise image authentication method is higher compared to other methods using fragile authentication method.

#### 3.1. Description of Symbol Definitions

This section defines and describes all the symbols used in this paper.

1.  $W$ : the weight of the original image
2.  $H$ : the height of the original image
3.  $I_O$ : the original image
4.  $I_W$ : the watermarked image
5.  $I_T$ : the tampered image
6.  $I_R$ : the recovered image
7.  $m \times n$ : the size of a block
8.  $N = (H \times W) / (m \times n)$ : the total number of the blocks in the image
9.  $B_i (i = 1, 2, \dots, N)$ : each block of the image
10.  $P_j^i (j = 1, 2, \dots, n \times m)$ : the pixel value of each block
11.  $SK$ : the secret key
12.  $rm_i$ : the pseudo-random number that is generated by  $SK$

13.  $BM_i$ : the mapping block
14.  $M_i$ : the mean value of each block
15.  $R_i$ : the recovery data of each block
16.  $A_i$ : the authentication data of each block
17.  $R'_i$ : the recovery data that is generated from  $I_T$
18.  $A'_i$ : the authentication data that is generated from  $I_T$
19.  $k$ : the size of the authentication data
20.  $q$ : the number of bits that will be embedded into each pixel
21.  $T_i$ : the table that is marked whether it has been tampered with
22.  $TM_i$ : the mapping table of  $T_i$
23.  $L$ : the enlarged image

### 3.2. Block-Wise Detection

First, the original image  $I_O$  with a size of  $H \times W$  is divided into  $N$  number of non-overlapping blocks  $B_i(i = 1, 2, \dots, N)$  of  $m \times n$  sizes. Recovery data,  $M_i$ , is the mean value of each block  $B_i$ , and the authentication data is created by encrypting the block information. The embedding process uses the least significant bit (LSB) replacement method to embed every  $q$  number of LSB pixel values in each block, which combines the  $k$ -bits authentication data  $A_i$  and recovery data for mapping block  $BM_i$ . The flowchart of watermark embedding process is shown in Figure 1. The following steps provide a detailed description on watermark generation and embedding process.

Step 1. The original image  $I_O$  is divided into  $N$  non-overlapping blocks  $B_i(i = 1, 2, \dots, N)$ . The size of each block  $B_i$  is  $m \times n$ .

Step 2. Using Equation (1) calculate the 8-bit mean value  $M_i$  of each block as this block's recovery data.

$$M_i = \text{round}\left(\left(\sum_{j=1}^{m \times n} P_j^i\right) / (m \times n)\right) \quad (1)$$

Step 3. The  $N$  pseudo-random numbers  $rm_i\{1, 2, \dots, N\}$ ,  $i = 1, 2, \dots, N$  are generated using the secret key  $SK$ . According to  $rm_i$ , the mapping block  $BM_i$ , that corresponds to block  $B_i$ , is created. The relationship between the original block  $B_i$  and the mapping block  $BM_i$  is shown in Equation (2). And the recovery data  $R_i$  of the block  $B_i$  is the mean value  $M_{rm_i}$  of the mapping block  $BM_i$ , as shown in Equation (3). Table 1 shows the relationship between the original block and the mapping block.

$$BM_i = B_{rm_i} \quad (2)$$

$$R_i = M_{rm_i} \quad (3)$$

Step 4. The  $k$ -bit authentication data  $A_i$  of each block  $B_i$  is generated by Equations (4) and (5).

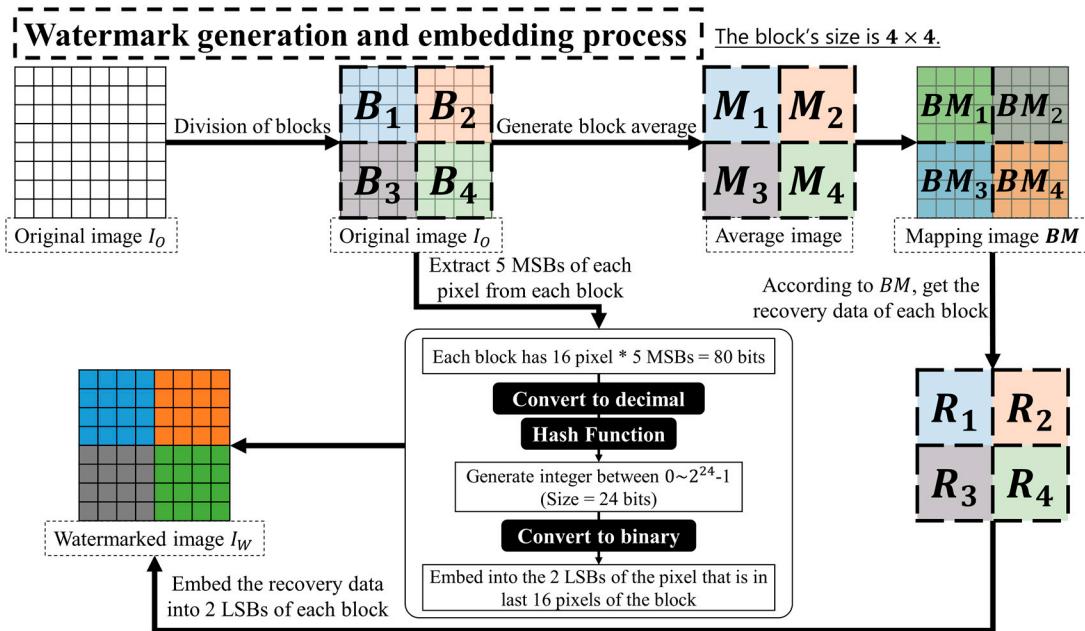
$$a_{5 \times (j-1)+x} = \text{mod}(P_j^i / 2^{8-(x-1)}, 2), x = (1, 2, \dots, 5) \quad (4)$$

$$A_i = \text{hash}\left(\sum_{x=1}^{m \times n \times 5} (a_x \times 2^{(x-1)})\right) \quad (5)$$

Step 5. The watermark  $W_i$  that will be embedded into each block  $B_i$  is generated using Equation (6).

$$W_i = (R_i \| A_i) \quad (6)$$

Step 6. Finally, the watermark  $W_i$  is embedded into the  $q$ -LSB of each block using the least significant bit (LSB) replacement method.



**Figure 1.** Flowchart of watermark generation and embedding process based on “block-wise detection” (The size of a block is  $4 \times 4$ ).

**Table 1.** Relationship between the original block and the mapping block.

i	1	2	3	4	...	N
<b>SK generation</b>	↓	↓	↓	↓	↓	↓
$rm_i$	N	4	1	2		3
$B_i$	$B_1$	$B_2$	$B_3$	$B_4$	...	$B_N$
<b>Associated blocks</b>	↓	↓	↓	↓	↓	↓
$B_{rm_i}$	$B_N$	$B_4$	$B_4$	$B_2$		$B_3$
$BM_i$	$BM_1$	$BM_2$	$BM_3$	$BM_4$		$BM_N$

In the image tamper detection process, steps of generating authentication data are similar to the steps mentioned in the section above. These steps are used to obtain  $k$ -bit authentication data  $A'_i$  for block  $B_i$ . The authentication data  $A_i$  that was embedded in block  $B_i$  is then extracted from the tampered image  $I_T$ . Comparing the similarity of the authentication data  $A_i$  that was embedded in Block  $B_i$ , to the re-calculated authentication data  $A'_i$ , if  $A'_i = A_i$ , indicates that the image was not tampered. On the other hand, if the extracted authentication data  $A_i$  differs from the re-calculated authentication  $A'_i$  ( $A'_i \neq A_i$ ), it indicates that the image block may have been tampered. We mark these image blocks as tampered areas. The flowchart of the tamper detection process is shown in Figure 2. The following steps provide a detailed description on tamper detection and image recovery methods.

Step 1. The tampered image  $I_T$  is divided into non-overlapping block  $B_i$ . Each block's size is  $m \times n$ .

Step 2. The re-calculated authentication data  $A'_i$  of the block  $B_i$  is generated by Equations (4) and (5) from  $I_T$ .

Step 3. The watermark  $W_i$  is extracted from the  $q$ -LSB of the block  $B_i$ 's pixel  $P_j^i$  in  $I_T$ . And using the function  $\text{RightSubBit}(W_i, k)$ , the extracted authentication data  $A_i$  is extracted from the right  $k$  bits of  $W_i$ .

Step 4. The re-calculated authentication data  $A'_i$  is compared with the extracted authentication data  $A_i$ . According to Equation (7), if the result is not the same, then  $T_i = 1$ , and it represents this block

as ‘a tampered block’. Otherwise, if the result is the same, then  $T_i = 0$ , and it represents this block as ‘not a tampered block’.

$$T_i = \begin{cases} 1, & A'_i \neq A_i \\ 0, & A'_i = A_i \end{cases} \quad (7)$$

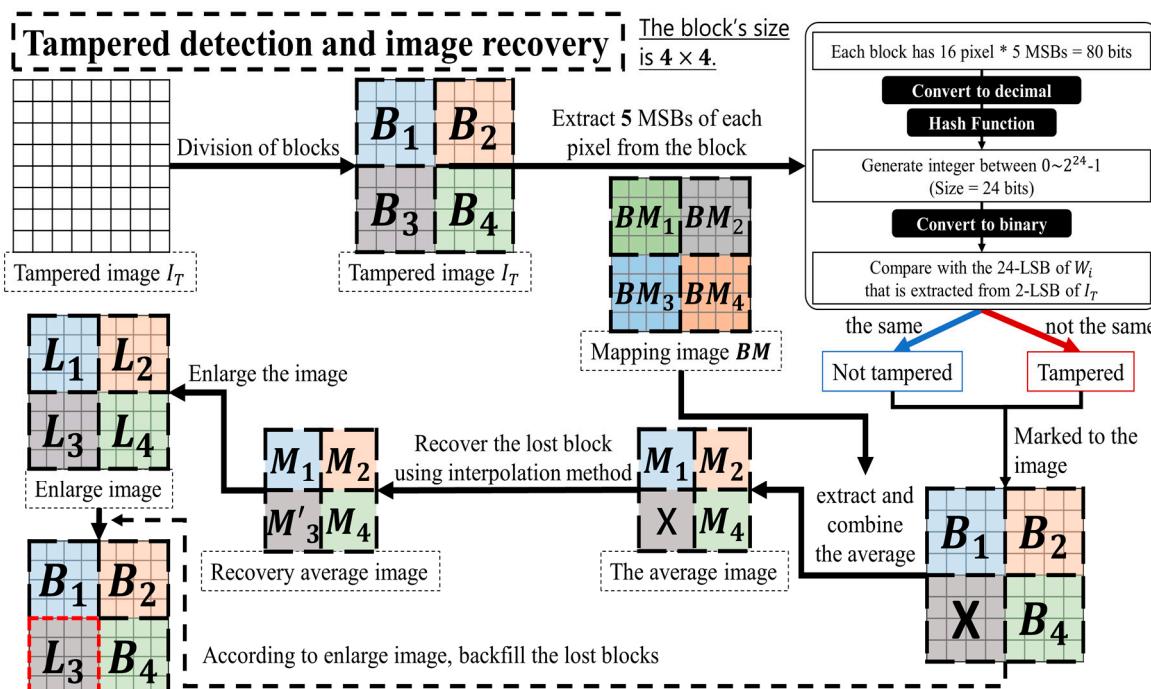
Step 5. The block  $B_i$  and the tamper mark table  $T_i$  generate the mapping block  $BM_i$  and the mapping tamper mark table  $TM_i$  using the secret key  $SK$ , respectively.

Step 6. For the mapping block  $BM_i$  that is  $TM_i = 0$ , the recovery data  $R_i$  is extracted from the 8-MSB of the watermark  $W_i$ .

Step 7. If the tampered image  $I_T$  is not tampered, then the extracted recovery data  $R_i$  is the  $1/(m \times n)$  multiple images of the original image  $I_O$ . But if the image is tampered, then the extracted recovery data image will have lost pixels. Because the image texture is coherent, we can use the surrounding pixel values to fill in the lost pixels value by interpolation method.

Step 8. The enlarged image  $L$  is generated by Bicubic Interpolation function [36].

Step 9. The lost block ( $T_i = 1$ ) is filled by the corresponding block of the enlarged image  $L$ , and then the recovery image  $I_R$  is generated.



**Figure 2.** Flowchart of tamper detection and image recovery process based on “block-wise detection” (The size of a block is  $4 \times 4$ ).

### 3.3. Pixel-Wise Detection

The block-wise detection method was introduced in the previous section. As the authentication data is based on blocks, when tampered pixels are detected, the entire block to which the pixel belongs will be marked as tampered. However, if authentication data is embedded based on pixels, then only the tampered pixels will be marked as tampered when detected. Therefore, when embedding authentication data, the pixel-wise image authentication method is much more accurate than those based on the block-wise image authentication method. This section describes a data hiding technique based on the pixel-wise image authentication method.

The process of generating a watermark is similar to the method based on block detection. The flow chart is shown in Figure 3 and the following steps provide a detailed description of the watermark generation and the embedding process:

Step 1. The original image  $I_O$  is divided into  $4 \times 4$  non-overlapping blocks  $B_i$ .

Step 2. According to Equation (1), the mean value  $M_i$  of each block  $B_i$  is calculated. The mean value of each block has two copies, as in Equation (8). The non-repeating pseudo-random sequence  $RM = \{rm_1, rm_2, \dots, rm_{N \times 2}\}$  is generated by the secret key  $SK$ . And two copies of the mean value are scrambled by Equation (9) as the recovery data  $R$ . The recovery data  $R_i$  which will be embedded into each block  $B_i$  is  $\begin{bmatrix} M_{rm_1} & M_{rm_{i+N}} \end{bmatrix}$ . The reason for embedding two copies of the block mean value in the image is that the survival chances of the recovery data may increase after the image has been maliciously tampered.

$$M_C = \begin{bmatrix} M_1 & M_1 \\ M_2 & M_2 \\ \vdots & \vdots \\ M_N & M_N \end{bmatrix} \quad (8)$$

$$R = \begin{bmatrix} M_{rm_1} & M_{rm_{N+1}} \\ M_{rm_2} & M_{rm_{N+2}} \\ \vdots & \vdots \\ M_{rm_N} & M_{rm_{N+N}} \end{bmatrix} \quad (9)$$

Step 3. According to Equation (10), the 8-bit encryption code  $H_j^i$  is generated by hash function after extracting 5 most significant bits (MSBs) of each pixel value  $P_j^i$ . Then the 8th least significant bit (LSB), the 7th LSB, the 2nd LSB, the 1st LSB, and the 6th LSB, the 5th LSB, the 4th LSB, the 3rd LSB of the encryption code  $H_j^i$  are calculated by exclusive or (XOR) operation, respectively. Thereafter, the authentication data  $A_j^i = \{a_1, a_2\}$  is generated.

$$\begin{cases} p_x = \text{mod}(P_j^i / 2^x, 2) \quad x = 1, 2, \dots, 5 \\ H_j^i = \text{hash}\left(\sum_{x=1}^5 (p_x \times 2^{(x-1)})\right) \end{cases} \quad (10)$$

Step 4. Finally, the recovery data  $R$  is embedded into 1st LSB of each pixel value in each block, and the authentication data  $\{a_1, a_2\}$  is embedded into the 2nd LSB and the 3rd LSB of the pixel, respectively.

In the tamper detection process, since authentication data is embedded based on pixels, the tampered pixels will be marked when any tamper is detected. Therefore, the result of tamper detection is much more accurate for the pixel-wise method compared to the block-wise method. The following steps provide a detailed description of tamper detection and self-recovery.

Step 1. Similar to the tamper detection and image recovery process described in Section 3.2, the re-calculation of authentication data  $A'^i_j$  is done from each pixel of the tampered image  $I_T$  with reference to Equation (10) and Figure 4.

Step 2. The authentication data  $A_j^i$  is extracted from the 2nd and the 3rd LSBs of each pixel  $P_j^i$ , and is compared with the re-calculated authentication data  $A'^i_j$ .

Step 3. If the extracted authentication data  $A_j^i$  is the same as the re-calculated authentication data  $A'^i_j$ , then the pixel is not tampered and  $T_i = 0$ . In other cases, the pixel is tampered and  $T_i = 1$ .

$$T_i = \begin{cases} 1, A'^i_j \neq A_j^i \\ 0, A'^i_j = A_j^i \end{cases} \quad (11)$$

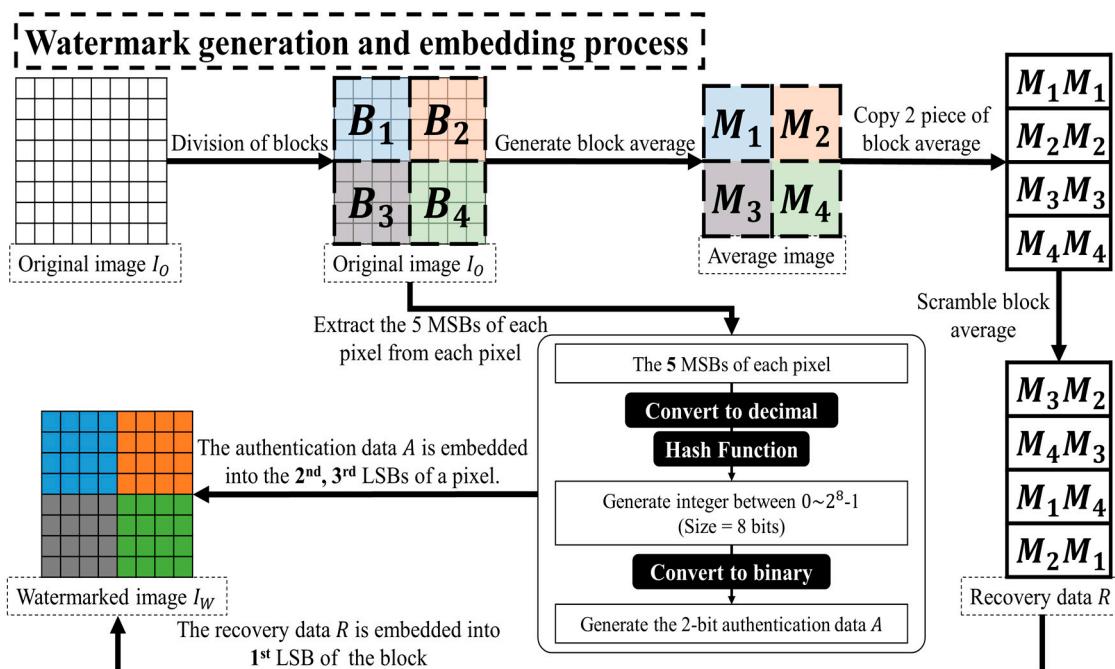
Step 4. First, the entire image is divided into non-overlapping blocks. For  $T_i = 0$ , the 1st LSB of each pixel in the block is extracted from the block of the non-tampered pixel, and combined as the recovery data  $R_i$  of that block.

Step 5. Since the recovery data  $R_i$  are the two copies of the block mean value that were disrupted, the recovery data is sorted using the secret key  $SK$ .

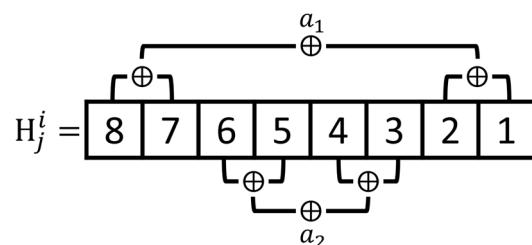
Step 6. If the number of the extracted mean value is less than half of the number of the original mean value, then skip to Step 8; otherwise, the tampered pixel is recovered from Step 7.

Step 7. The recovery data  $R_i$  is combined as a smaller image, and the lost pixel value is filled by the interpolation method. The tampered pixel is corrected to the pixel of the smaller image.

Step 8. If the recovery data is incomplete, the tampered pixel cannot be recovered. So the mean value of the surrounding eight pixel values is calculated, and the tampered pixel is corrected to this value.

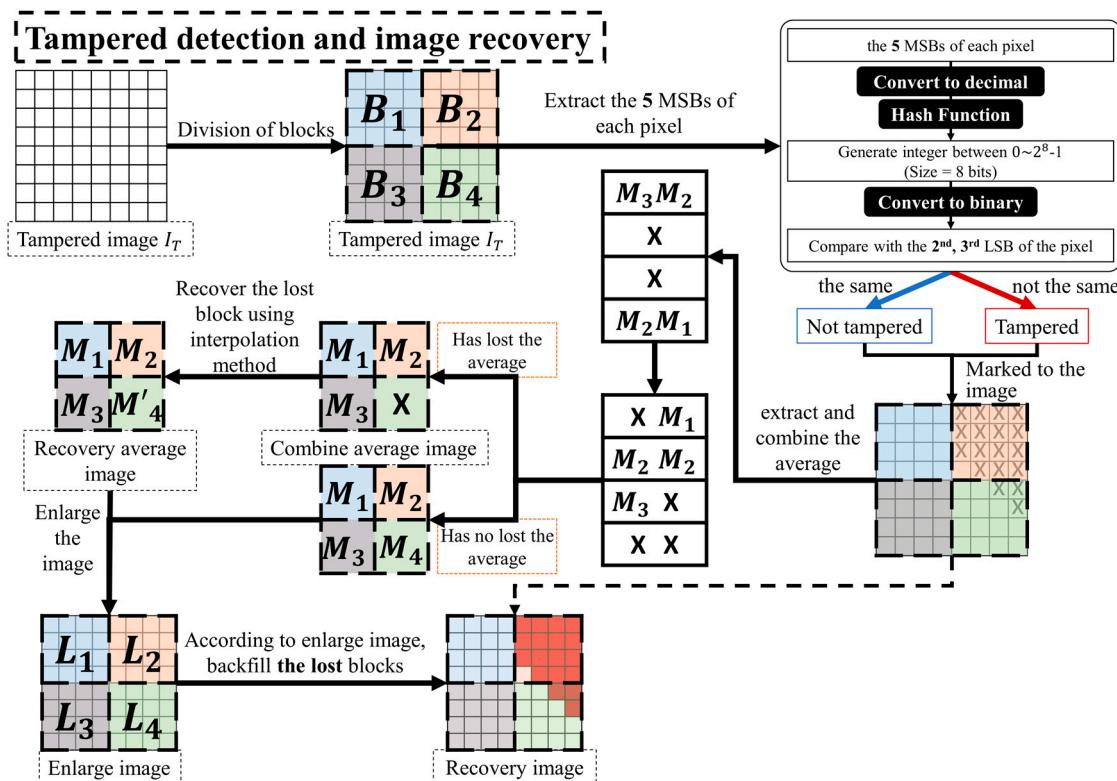


**Figure 3.** Flowchart of the watermark generation and embedding process based on “pixel-wise detection”.



**Figure 4.** The authentication data  $A = \{a_1, a_2\}$  generation.

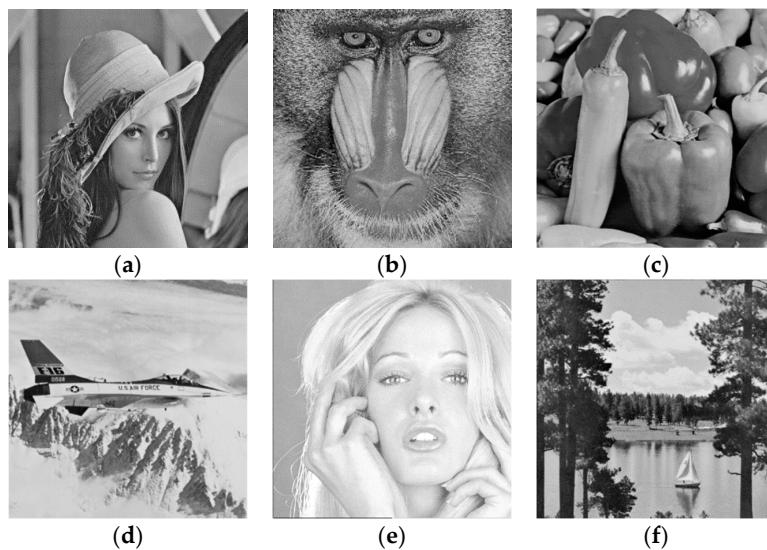
Figure 5 displays the flowchart of tamper detection and image recovery process based on “pixel-wise detection.”



**Figure 5.** Flowchart of tamper detection and image recovery process based on “pixel-wise detection”.

#### 4. Experimental Results and Comparison

The experimental results of this paper are performed using MATLAB 2017a in a Windows 10, Intel Core i7 3.60 GHz system with 4 GB of memory. We used six grayscale images of size  $512 \times 512$  (Lena, Baboon, Peppers, Airplane, Tiffany, Lake) as shown in the Figure 6.



**Figure 6.** Six  $512 \times 512$  test images. (a) Lena, (b) Baboon, (c) Peppers, (d) Airplane, (e) Tiffany, (f) Lake.

For the measurement results of tampered detection, we evaluated the error rate of tampered detection by False Negative Rate (FNR) and False Positive Rate (FPR). The algorithm is shown in Equations (12) and (13). The error of tampered detection is divided into False Negative (FN) and False Positive (FP). The False Positive is the number of the pixel that is a non-tampered pixel but is

detected as tampered; the False Negative is the number of pixels that are tampered but are detected as a non-tampered pixel. And TP (True Positive) denotes the number that are correctly detected for the tampered pixel, and TN (True Negative) denotes the number that are correctly detected for the untampered pixel [35].

$$FNR = FN \div (FN + TP) \quad (12)$$

$$FPR = FP \div (FP + TN) \quad (13)$$

The image quality measurement of the watermarked image and the recovered image uses peak signal to noise ratio (PSNR). As a result, if the image has a higher PSNR, the image quality is better. Their equations are represented as Equations (14) and (15), where  $H$  and  $W$  are the height and width of the image, and  $I_O(i)$  and  $I_T(i)$  are the  $i$ th pixel of the original image and the measured image.

$$MSE = \frac{1}{H \times W} \sum_{i=1}^{H \times W} (I_O(i) - I_T(i))^2 \quad (14)$$

$$PSNR = 10 \times \log \frac{255^2}{MSE} \quad (15)$$

In addition, we also use the Structural Similarity Index (SSIM) to measure the similarity between the original image and the watermarked image. The higher SSIM value indicates that the similarity between the original image and the watermark image is higher. The calculation method is shown in Equation (16), where  $\mu_x$  and  $\mu_y$  are the average values of the original image and the watermarked image,  $\sigma_{xy}$  is the co-variation of the original image and the watermarked image, and  $\sigma_x$  and  $\sigma_y$  are the variation of the original image and the watermarked image, respectively.  $C_1$  and  $C_2$  are constants.

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (16)$$

#### 4.1. Digital Image Tamper Detection

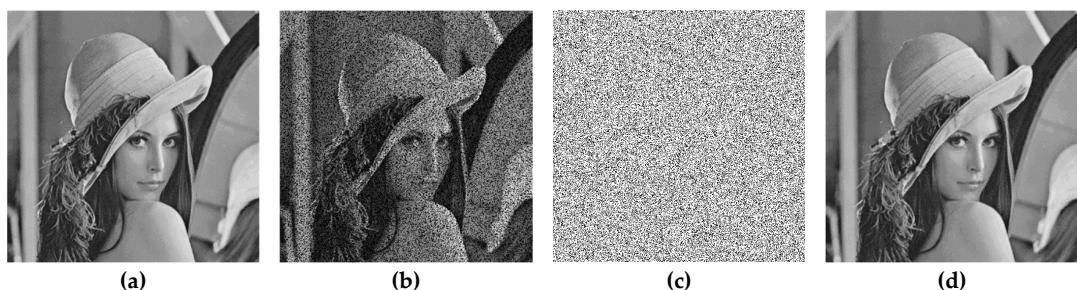
This section analyzes and explores embedding methods utilizing different block sizes. For the block-wise detection method, we used three different block sizes of  $2 \times 4$ ,  $3 \times 3$ , and  $4 \times 4$ , to examine and compare error rates as well as image quality. The results obtained are as shown in Table 2. The tamper rate is set at 10%. The condition of  $q = 3$ ,  $k = 16$  is set for the  $2 \times 4$  block size, which indicates that the watermark is embedded in the 3 LSBs of each pixel and the authentication data length is 16-bit. The false positive rate (FPR) is 0.17335% and the image quality of the watermarked and recovered images are  $PSNR^{(w)} = 41.28$  dB and  $PSNR^{(r)} = 44.48$  dB, respectively. Under the condition of  $q = 3$ ,  $k = 16$ , the FPR is 0.13257% for  $3 \times 3$  block size, which is lower than the FPR for  $2 \times 4$  block size. The difference in the block size will affect FPR, as the entire block will be marked as a tampered block when a tampered pixel is detected using this method. The watermark image quality  $PSNR^{(w)}$  and recovered image quality  $PSNR^{(r)}$  for images divided into  $3 \times 3$  blocks are slightly better than that of images divided into  $2 \times 4$  blocks. Lastly, when the block size is  $4 \times 4$  and the condition is set to  $q = 2$ ,  $k = 24$ , the  $PSNR^{(w)} = 47.32$  dB, which is much higher than the other two block sizes; and  $PSNR^{(r)} = 44.68$  dB, which is slightly higher when compared to the other two block sizes. This may be because the watermark is only embedded in the 2 LSBs of a pixel. Overall, the quality of the recovered images  $PSNR^{(r)}$  are similar to each other, which is about 44 dB.

Figure 7 shows the results of tamper detection and recovery from salt and pepper noise attack. Figure 7c is the tamper detection result, which shows no error in detection. Figure 7d is the result of the recovered image, which shows that its image quality,  $PSNR^{(r)}$ , is 40.68 dB. In Figure 8, a flower is added on top of Lena's hat, and the block-wise detection method was used. Figure 8a–c shows tampered images with a tamper rate of 4.4%. Figure 8d–f shows the detection results, and Figure 8g–i shows

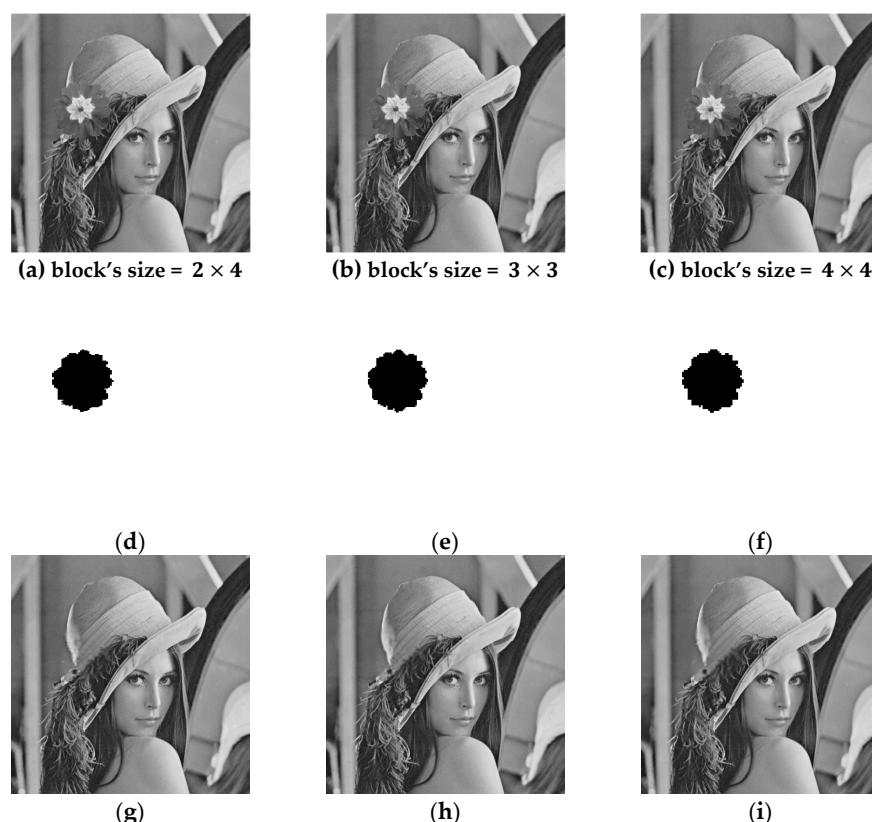
recovered images. The quality of recovered images is all above 45 dB. Figure 9 indicates the tamper detection results using  $4 \times 4$  blocks. Figure 9a–d are tampered images of Lena, Baboon, Airplane, and Lake, with a tamper rate of 3.7%. Figure 9e–h are the results of image recovery, and the image quality  $\text{PSNR}^{(r)}$  are all above 45 dB. It is evident that our proposed image authentication method could resist diverse tampering attacks.

**Table 2.** Comparison of the average error and image quality for block-wise detection.

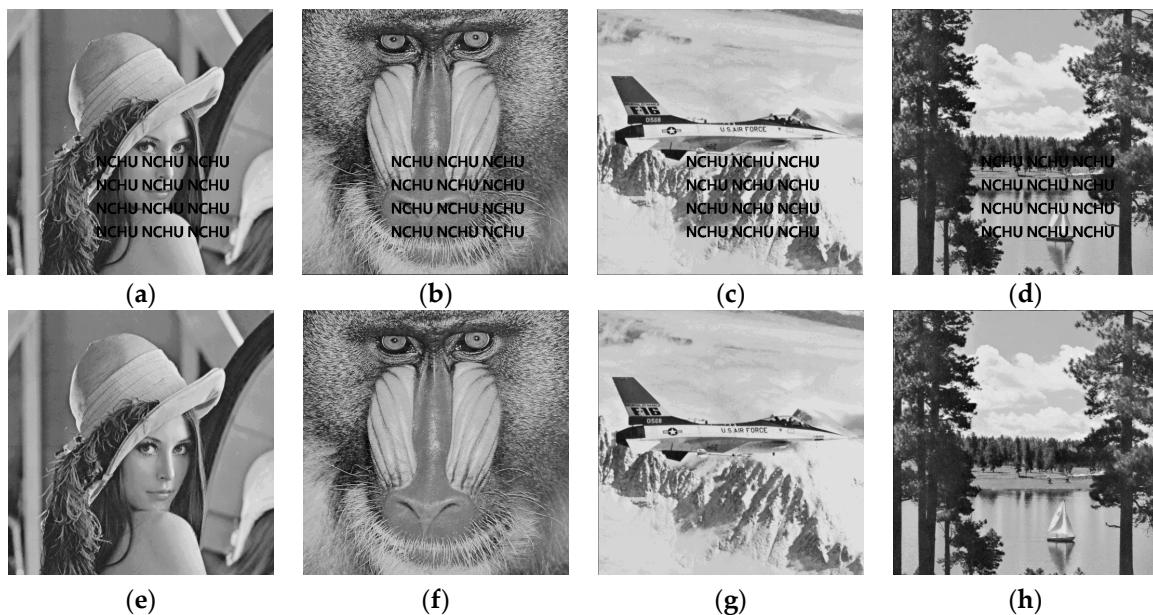
$m$	$n$	$q$	$k$	FPR (%)	$\text{PSNR}^{(w)}$ (dB)	$\text{PSNR}^{(r)}$ (dB)
2	4	3	16	0.17335	41.28	44.48
3	3	3	16	0.13257	41.77	44.51
4	4	2	24	0.17335	47.32	44.68



**Figure 7.** Watermarked Lena and tamper detection results using  $4 \times 4$  pixel-wise detection  
(a) Watermarked image ( $\text{PSNR}^{(w)} = 41$  dB), (b) tampered image (Tamper rate = 30 %), (c) tamper detection result (FPR = 0%), (d) recovery image ( $\text{PSNR}^{(r)} = 40.68$  dB ).



**Figure 8.** Watermarked Lena and tamper detection results for different block sizes using “block-size detection” (a–c) Tampered images (tamper rate = 4.4%), (d–f) tamper detection result, (g–i) recovery images.



**Figure 9.** Tamper detection results using “pixel-wise detection” (The test images are Lena, Baboon, Airplane and Lake) (a–d) Tampered images (tamper rate = 3.7%), (e–h) recovery images.

The block-wise detection method that we proposed uses a  $4 \times 4$  block size. Each image’s PSNR and error rate for different tamper rates are shown in Table 3. Since watermark authentication data are embedded in the 2 LSBs of a pixel, the  $\text{PSNR}^{(w)}$  of every image is higher than 47 dB. When the tamper rate is 10%, the  $\text{PSNR}^{(r)}$  of the recovered image is close to 50 dB for Lena’s image. At a high tamper rate of 50%, the recovered image  $\text{PSNR}^{(r)}$  of each image is still above 32 dB. Airplane and Tiffany images are the best performing images with a  $\text{PSNR}^{(r)} = 36.79$  dB. With respect to error rate, since our method uses blocks to detect tampered blocks, the error rate is lesser subjected to the effects of higher tamper rates when clipping attacks occur.

**Table 3.** Comparison of the  $4 \times 4$  “block-wise detection”.

Image	PSNR <sup>(w)</sup> (dB)	Tamper Rate	10%	20%	30%	40%	50%
Lena	47.20	PSNR <sup>(r)</sup> (dB)	49.47	44.39	41.23	38.58	36.61
		FNR (%)	0	0	0	0	0
		FPR (%)	0.173	0.391	0.669	0.787	0
Baboon	47.29	PSNR <sup>(r)</sup> (dB)	38.69	35.55	33.95	32.93	32.13
		FNR (%)	0	0	0	0	0
		FPR (%)	0.173	0.391	0.669	0.787	0
Peppers	47.23	PSNR <sup>(r)</sup> (dB)	42.84	40.54	38.32	36.76	35.17
		FNR (%)	0	0	0	0	0
		FPR (%)	0.173	0.391	0.669	0.787	0
Airplane	47.33	PSNR <sup>(r)</sup> (dB)	46.59	44.54	42.83	40.32	36.79
		FNR (%)	0	0	0	0	0
		FPR (%)	0.173	0.391	0.669	0.787	0
Tiffany	47.54	PSNR <sup>(r)</sup> (dB)	45.81	41.87	40.08	38.36	36.79
		FNR (%)	0	0	0	0	0
		FPR (%)	0.173	0.391	0.669	0.787	0

#### 4.2. Comparison with Other Methods

Table 4 shows a comparison of efficiency between our method and [21,22,24,31,32] for maximum tampering rate using the Lena image. For the methods proposed by Yang and Shen [21], Yang et al. [22],

and Kim et al. [32], the maximum tolerable tamper rate is less than 50%, which is the same as our methods. But the PSNR<sup>(r)</sup> values for block-wise method and pixel-wise method of our methods are 36.62 dB and 37.26 dB, which is higher than those of Yang and Shen's [21], Yang et al.'s [22] and Kim et al.'s [32] methods. In addition, the PSNR<sup>(r)</sup> of Qin et al. [31] is higher than 3.74 dB compared to our method based on pixel-wise detection method, but their maximum tolerable tamper rate is less than 45%. Table 5 shows the SSIM performance of the proposed method using six standard testing images. The results show that the block-wise method has better structure similarity for image Baboon and least similarity for image Tiffany. As for the pixel-wise method, image Baboon still has better similarity, and image Peppers has the least similarity when using the pixel-wise method. Overall, the 4 × 4 block-wise method has the best SSIM performance at 0.986461 and the worst SSIM at 0.955061. Table 6 shows a comparison of the false positive rate (FPR) and the false negative rate (FNR) of the proposed methods and the previous methods [25,26,28,34] using several test images.

**Table 4.** Performance comparisons of proposed methods and [21,22,24,31,32].

Methods	PSNR <sup>(w)</sup> (dB)	PSNR <sup>(r)</sup> (dB)	Condition of Restoration
Yang and Shen [21]	40.7	32	<50%
Yang et al. [22]	51.3	36	<50%
Qian et al. [24]	37.9	35	<35%
Qin et al. [31]	46	41	<45%
Kim et al. [32]	43.7	33.6	<50%
Proposed method (4 × 4 block-wise detection)	43.73	36.62	<50%
Proposed method (4 × 4 pixel-wise detection)	41	37.26	<50%

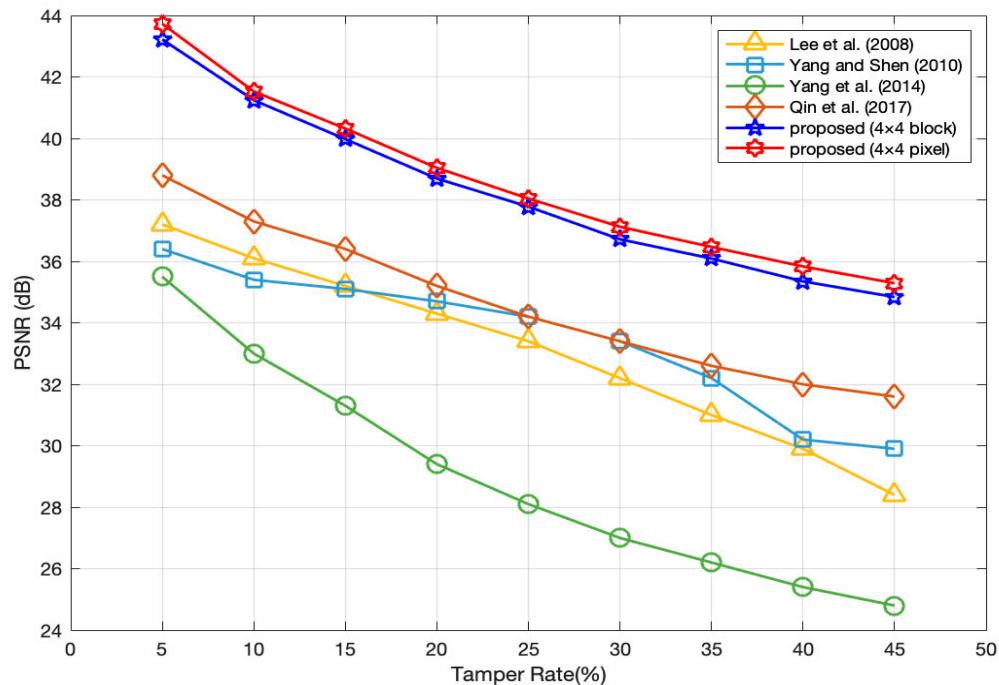
**Table 5.** Structural Similarity Index (SSIM) performance of proposed methods.

	2 × 4 Block-Wise	3 × 3 Block-Wise	4 × 4 Block-Wise	4 × 4 Pixel-Wise
512_lena	0.92739	0.93511	0.958791	0.94159
512_baboon	<b>0.97581</b>	<b>0.97843</b>	<b>0.986461</b>	<b>0.97489</b>
512_peppers	0.93037	0.93739	0.961136	<b>0.93938</b>
512_airplane	0.92725	0.93173	0.95682	0.94193
512_tiffany	<b>0.91812</b>	<b>0.92558</b>	<b>0.955061</b>	0.94091
512_lake	0.94736	0.95261	0.970439	0.96453

**Table 6.** False positive rate (FPR) and false negative rate (FNR) comparison of proposed methods and [25,26,28,34].

Methods	FPR(%)	FNR(%)
Tong et al. [25]	0.22	0
Chen et al. [26]	0.25	0
Ansari et al. [28]	0.5	0.01
Wang et al. [34]	0.30	0
Proposed method (4 × 4 block-wise detection)	0.173	0
Proposed method (4 × 4 pixel-wise detection)	0	0

The comparison of recovered image quality are shown in Figure 10 under different tamper rates among our proposed methods and other fragile watermarking methods such as Lee et al. [20], Yang and Shen [21], Yang et al. [22], Qin et al. [31]. From this, it is evident that for our method, the PSNR of recovered images is about 2 dB higher than that of other methods at any tamper rate. Therefore, in terms of image recovery, the performance of our method is excellent.



**Figure 10.** Peak signal to noise ratio ( $\text{PSNR}^{(r)}$ ) comparison of the recovered image between the proposed methods and methods in [20–22,31].

Table 7 shows the performance of the proposed method in terms of FPR, FNR, and recovered image quality when dealing with complex images. Baboon was used to represent these kinds of image attacks. According to Table 7, even if 14.35% of Baboon was tampered its recovered PSNR is still above 38 dB, and with  $4 \times 4$  block size the PSNR by pixel-wise approach gets even higher to 39.21 dB. Furthermore, after examining several complex images, the FNR was always found to be 0, which means that the area which has been tampered will always be detected.

**Table 7.** Recovered  $\text{PSNR}^{(r)}$ , FPR, and FNR results of the proposed method with complex images.

	$2 \times 4$ Block-Wise	$3 \times 3$ Block-Wise	$4 \times 4$ Block-Wise	$4 \times 4$ Pixel-Wise
<b>Tamper Rate (%)</b>	14.35	14.35	14.35	14.35
<b>Recovered <math>\text{PSNR}^{(r)}</math> (dB)</b>	38.749796	38.855165	38.192091	39.2149
<b>FNR(%)</b>	0	0	0	0
<b>FPR(%)</b>	0.029	0.025	0.038	0

Table 8 shows the results of 500 test images of the size  $512 \times 512$  using the proposed block-wise method. It can be seen that the FPR and FNR values become lower or even approached to zero when the block size reduces. The highest values of FNR and FPR were always below 0.000082 and 0.038, respectively. In addition, the recovered  $\text{PSNR}_R$  value was always above 36 dB.

**Table 8.** Recovered PSNR, FPR, and FNR results of the proposed method (block-wise) using 500 test images under the tamper rate of 14.35% with variant block sizes.

Block Size		$2 \times 4$	$3 \times 3$	$4 \times 4$
Tamper Rate (%)		14.35	14.35	14.35
Recovered PSNR <sup>(tr)</sup>	average	41.55	41.54	40.61
	highest	<b>47.55</b>	<b>48.09</b>	<b>48.18</b>
	lowest	37.45	37.41	36.77
FNR(%)	average	0.000082	0.000274	0.000562
	highest	0.000824	0.003803	0.006887
	lowest	<b>0.000000</b>	<b>0.000000</b>	<b>0.000000</b>
FPR(%)	average	0.02850	0.02531	0.03764
	highest	0.02855	0.02548	0.03803
	lowest	<b>0.02801</b>	<b>0.02324</b>	<b>0.03526</b>

Table 9 shows the comparison of the processing complexity of each method. Images of size  $256 \times 256$  were used to examine each method. These watermarking schemes contain block types, authentication information, and recovery information. Each method's processing steps are displayed in terms of block division, discrete wavelet transform (DWT), DWT embedding, discrete cosine transform (DCT), mean calculation, watermark hashing, Hilbert curve transformation, exclusive or (XOR), least significant bit (LSB) embedding, and histogram shifting and modification. The total number of steps in the watermarking procedure are also presented for each method. In Table 9, there will be higher operation counts for pixel-wise methods, such as Lo and Hu [27], Qin et al. [31] and the proposed pixel-wise method. The methods of Hsu and Tu [23], Yin et al. [30], Singh and Singh [29], Tai and Liao [33] are block-wise. Among them, the proposed block-wise method has the least operation count.

**Table 9.** Processing complexity comparison.

	Hsu and Tu [23]	Lo and Hu [27]	Singh & Singh [29]	Yin et al. [30]	Qin et al. [31]	Tai and Liao [33]	Proposed Block-Wise	Proposed Pixel-Wise
Block size	$8 \times 8$	$4 \times 4$	$2 \times 2$	$4 \times 4$	$3 \times 3$	$4 \times 4$	$4 \times 4$	$4 \times 4$
Block division	5120	4096	16,384	4096	64,516	4096	4096	4096
Predictive coding and it inverse	0	65,536	0	0	0	0	0	0
Arnold's permutation	0	0	0	0	0	4096	0	0
DWT	4096	0	0	16,384	0	4096	0	0
DWT embedding	0	0	0	0	0	4096	0	0
DCT	0	0	16,384	0	0	0	0	0
Mean calculation	4096	0	16,384	0	0	0	4096	4096
Watermark hashing	1024	0	16,384	0	64,516	4096	4096	65,536
Hilbert curve transformation	1024	0	0	4096	64,516	0	0	0
XOR	20,480	0	0	4096	0	4096	4096	65,536
LSB embedding	1024	0	16,384	0	0	0	4096	65,536
Histogram shifting and modification	1024	65,536	16,384	4096	64,516	0	0	0
Total operation count	37,888	139,264	81,920	32,768	258,064	62,464	20,480	204,800

## 5. Conclusions

In this paper, we proposed a self-recovery fragile watermarking image authentication technology for wireless sensor networks. The authentication methods are classified into two types: block-wise and pixel-wise methods. In the block-wise detection method, authentication data is generated from each block, and the average block value is used to generate recovery data. Further, the length of authentication data and recovery data are adjusted according to the size of each block. From the experimental results, it can be seen that when a tampered pixel is detected, the block to which it belongs, will be marked as a tampered block. Therefore, if the block division is small, the false positive rate (FPR) will be small as well. In the pixel-wise detection method, the authentication data is generated

from each pixel, and the recovery data is generated from the mean value of the  $4 \times 4$  block. When the tamper rate is 50%, the PSNR of the recovered image of all tested images is above 32 dB. Compared with other methods, our method has a better performance for tamper detection and image recovery.

Using our proposed method, future research should attempt to use different extraction techniques that demonstrate characteristics of the block during the process of generating recovery data, such as the absolute moment block truncation coding (AMBTC) method. In addition to the common clipping and peppered salt attacks, we did not conduct relevant vector quantization attack experiments with respect to tamper experiments. Future research can consider them for in-depth studies. In addition, our method belongs to the fragile watermarking method. After using common image processes, such as JPEG compression, the recovery and authentication data embedded in the image get destroyed, and tampered areas cannot be detected accurately. But JPEG compression can reduce the need of transmitting bandwidth effectivity in WSNs. Therefore, authentication robustness in image compression is also a topic of concern which can be explored in further studies.

**Author Contributions:** Conceptualization, C.-F.L. and J.-J.S.; Methodology, C.-F.L., J.-J.S., Z.-R.C.; Software, C.-F.L., Z.-R.C.; Validation, C.-F.L., Z.-R.C.; Formal Analysis, C.-F.L., J.-J.S.; Investigation, C.-F.L., Z.-R.C.; Resources, C.-F.L., J.-J.S.; Data Curation, C.-F.L., J.-J.S.; Writing—Original Draft Preparation, C.-F.L., Z.-R.C.; Writing—Review & Editing, Somya Agrawal; Visualization, Z.-R.C., S.A.; Supervision, C.-F.L., J.-J.S.; Project Administration, C.-F.L., J.-J.S.; Funding Acquisition, C.-F.L.

**Acknowledgments:** This research was partially supported by the Ministry of Science and Technology of the Republic of China under the Grants MOST 106-2221-E-324-006 -MY2.

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

## References

1. Feng, J.; Potkonjak, M. Real-time watermarking techniques for sensor networks. *Proc. SPIE* **2003**. [[CrossRef](#)]
2. Wong, J.L.; Feng, J.; Kirovski, D.; Potkonjak, M. Security in sensor networks: Watermarking techniques. In *Wireless Sensor Networks*; Raghavendra, C.S., Sivalingam, K.M., Znati, T., Eds.; Springer: Boston, MA, USA, 2004.
3. Shi, X.; Xiao, D. A reversible watermarking authentication scheme for wireless sensor networks. *Inf. Sci.* **2013**, *240*, 173–183. [[CrossRef](#)]
4. Singh, A.K.; Kumar, B.; Dave, M.; Mohan, A. Robust and imperceptible dual watermarking for telemedicine applications. *Wirel. Pers. Commun.* **2015**, *80*, 1415–1433. [[CrossRef](#)]
5. Lou, D.-C.; Liu, J.-L. Fault resilient and compression tolerant digital signature for image authentication. *IEEE Trans. Consum. Electron.* **2000**, *46*, 31–39.
6. Tsai, P.; Hu, Y.; Chang, C. Novel image authentication scheme based on quadtree segmentation. *Imaging Sci. J.* **2005**, *53*, 149–162. [[CrossRef](#)]
7. Ababneh, S.; Ansari, R.; Khokhar, A. Iterative compensation schemes for multimedia content authentication. *J. Vis. Commun. Image Represent.* **2009**, *20*, 303–311. [[CrossRef](#)]
8. Umamageswari, A.; Suresh, G.R. Secure medical image communication using ROI based lossless watermarking and novel digital signature. *J. Eng. Res.* **2014**, *2*, 87–108. [[CrossRef](#)]
9. Das, C.; Panigrahi, S.; Sharma, V.K.; Mahapatra, K.K. A novel blind robust image watermarking in DCT domain using inter-block coefficient correlation. *AEU Int. J. Electron. Commun.* **2014**, *68*, 244–253. [[CrossRef](#)]
10. Mishra, A.; Agarwal, C.; Sharma, A.; Bedi, P. Optimized gray-scale image watermarking using DWT-SVD and firefly algorithm. *Expert Syst. Appl.* **2014**, *41*, 7858–7867. [[CrossRef](#)]
11. Parah, S.A.; Sheikh, J.A.; Loan, N.A.; Bhat, G.M. Robust and blind watermarking technique in DCT domain using inter-block coefficient differencing. *Digit. Signal Process.* **2016**, *53*, 11–24. [[CrossRef](#)]
12. Di, Y.-F.; Lee, C.-F.; Wang, Z.-H.; Chang, C.-C.; Li, J. A robust and removable watermarking scheme using singular value decomposition. *KSII Trans. Internet Inf. Syst.* **2016**, *12*, 5268–5285.
13. Singh, A.K. Improved hybrid algorithm for robust and imperceptible multiple watermarking using digital images. *Multimed. Tools Appl.* **2017**, *76*, 8881–8900. [[CrossRef](#)]

14. Huynh-The, T.; Hua, C.-H.; Anh Tu, N.; Hur, T.; Bang, J.; Kim, D.; Bilal Amin, M.; Kang, B.H.; Seung, H.; Lee, S. Selective bit embedding scheme for robust blind color image watermarking. *Inf. Sci.* **2018**, *426*, 1–18. [[CrossRef](#)]
15. Zear, A.; Singh, A.K.; Kumar, P. A proposed secure multiple watermarking technique based on dwt, DCT and SVD for application in medicine. *Multimed. Tools Appl.* **2018**, *77*, 4863–4882. [[CrossRef](#)]
16. Lee, C.-F.; Chen, H.-L.; Yang, T.-C. Semi-blind watermarking scheme exploiting self-reference image. *J. Internet Technol.* **2011**, *12*, 313–326.
17. Preda, R.O. Semi-fragile watermarking for image authentication with sensitive tamper localization in the wavelet domain. *Measurement.* **2013**, *46*, 367–373. [[CrossRef](#)]
18. Al-Otum, H.M. Semi-fragile watermarking for grayscale image authentication and tamper detection based on an adjusted expanded-bit multiscale quantization-based technique. *J. Vis. Commun. Image Represent.* **2014**, *25*, 1064–1081. [[CrossRef](#)]
19. Qi, X.J.; Xin, X. A singular-value-based semi-fragile watermarking scheme for image content authentication with tamper localization. *J. Vis. Commun. Image Represent.* **2015**, *30*, 312–327. [[CrossRef](#)]
20. Lee, T.-Y.; Lin, S.D. Dual watermark for image tamper detection and recovery. *Pattern Recogn.* **2008**, *41*, 3497–3506. [[CrossRef](#)]
21. Yang, C.-W.; Shen, J.-J. Recover the tampered image based on VQ indexing. *Signal. Process.* **2010**, *90*, 331–343. [[CrossRef](#)]
22. 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 Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2014), Kitakyushu, Japan, 27–29 August 2014; pp. 130–133.
23. Hsu, C.-S.; Tu, S.-F. Image tamper detection and recovery using adaptive embedding rules. *Measurement* **2016**, *88*, 287–296. [[CrossRef](#)]
24. Qian, Z.; Feng, G.; Zhang, X.; Wang, S. Image self-embedding with high-quality restoration capability. *Digit. Signal Process.* **2011**, *21*, 278–286. [[CrossRef](#)]
25. Tong, X.J.; Liu, Y.; Zhang, M.; Chen, Y. A novel chaos-based fragile watermarking for image tampering detection and self-recovery. *Signal Process. Image Commun.* **2013**, *28*, 301–308. [[CrossRef](#)]
26. Chen, F.; He, H.J.; Tai, H.M.; Wang, H.X. Chaos-based self-embedding fragile watermarking with flexible watermark payload. *Multimed. Tools Appl.* **2014**, *72*, 41–56. [[CrossRef](#)]
27. Lo, C.-C.; Hu, Y.-C. A novel reversible image authentication scheme for digital images. *Signal Process.* **2014**, *98*, 174–185. [[CrossRef](#)]
28. Ansari, I.A.; Pant, M.; Ahn, C.W. SVD based fragile watermarking scheme for tamper localization and self-recovery. *Int. J. Mach. Learn. Cybern.* **2016**, *7*, 1225–1239. [[CrossRef](#)]
29. Singh, D.; Singh, S.K. Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. *J. Vis. Commun. Image Represent.* **2016**, *38*, 775–789. [[CrossRef](#)]
30. Yin, Z.; Niu, X.; Zhou, Z.; Tang, J.; Luo, B. Improved reversible image authentication scheme. *Cogn. Comput.* **2016**, *8*, 890–899. [[CrossRef](#)]
31. Qin, C.; Zhang, P.; Ji, X.; Dong, J.; Wang, J. Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. *Signal Process.* **2017**, *138*, 280–293. [[CrossRef](#)]
32. Kim, C.; Shin, D.; Yang, C.-N. Self-embedding fragile watermarking scheme to restoration of a tampered image using AMBTC. *Pers. Ubiquit. Comput.* **2018**, *22*, 11–22. [[CrossRef](#)]
33. Tai, W.-L.; Liao, Z.-J. Image self-recovery with watermark self-embedding. *Signal Process.* **2018**, *65*, 11–25. [[CrossRef](#)]
34. Wang, C.; Zhang, H.; Zhou, X. A Self-recovery fragile image watermarking with variable watermark capacity. *Appl. Sci.* **2018**, *8*, 548. [[CrossRef](#)]
35. Katzenbeisser, S.; Petitcolas, F.A.P. *Information Hiding Techniques for Steganography and Digital Watermarking* (Artech House Computer Security Series); Artech House: Norwood, MA, USA, 2000.
36. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 1153–1160. [[CrossRef](#)]

