Hierarchical Watermarking Scheme for Image Authentication and Recovery

Phen-Lan Lin¹
Dept. of Computer Science and Information Management
Providence University
Shalu, Taichung, Taiwan
lan@pu.edu.tw

Chung-Kai Hsieh², Po-Whei Huang³
Department of Computer Science
National Chung Hsing University
Taihphen, Taichung, Taiwan

Abstract
In this paper, we present an efficient and effective digital watermarking method for image tamper protection and recovery. Our method is efficient as it only uses simple operations such as parity check and the comparison between the average intensities. It is effective because the detection is hierarchical structured such that the accuracy of tamper localization can be ensured. That is, if a tampered block, as small as 2×2 pixels, is not detected in level-1 inspection, it will be detected in level-2 or level-3 inspection with probability nearly 1. Our method is also very storage effective, as it only requires a secret key K and a public chaotic mixing algorithm to recover a tampered image. Compare with the method in [8], our method not only is as simple and as effective in tamper detection and localization, it also provides with capability of tamper recovery.

1. Introduction
Because digital data can be edited easily and imperceptibly, as a result, content authenticity becomes greatly threatened. Classical image authentication mechanisms [1-2], called tamper-proofing mechanisms sometimes, attach a signature to the image and verify the contents by comparing the signature computed from the received image to the attached signature. These mechanisms can detect whether or not an image has been changed; however, they cannot identify how and where the image has been changed. Besides, the attached signature requires additional bandwidth or storage that may not always be available. To solve these problems, many researchers have proposed watermarking for image authentication [3-6].

Wong [5] presented a public-key fragile watermarking that embeds a digital signature of the MSBs of the block of the image into the LSBs of the same block. However, the scheme was soon attacked by Holliman and Memon with a counterfeit constructed from a vector quantization codebook generated from the image [9]. Following the Holliman and Memon's VQ codebook attack, several enhanced algorithms based on Wong's scheme were proposed [4][9][10]. Nonetheless, "they are either fail to effectively address the problem or sacrifice tamper localization accuracy of the original methods", as pointed out by Celik et al. [8]. They then presented an algorithm based on Wong's scheme and demonstrated that their algorithm can thwart the VQ codebook attack while sustaining the localization property [8].

In this paper, we present an efficient and effective digital watermarking method for image tamper protection and recovery. Our method is efficient as it only uses simple operations such as parity check and the comparison between the average intensities. It is effective because the detection is hierarchical structured and the block size increases as the hierarchical level increases such that the accuracy of tamper localization can be ensured. That is, if a tampered block, as small as 2×2 pixels, is not detected in level-1 inspection, it will be detected in level-2 or level-3 inspection with probability nearly 1. Our method is also very storage effective, as it only requires a secret key K and a public chaotic mixing algorithm to recover a tampered image. The experimental results demonstrate the precision of tamper detection and localization is 99.6% and 100% after level-2 and level-3 inspection. The tamper recovery rate is better than 93% for a 50% or less tampered image. Compare with the method in [8], our method not only is as simple and as effective in tamper detection and localization, it also provides with capability of tamper recovery.

2. The Proposed Scheme
2.1. Block-based Watermark Embedding
First, we divide the image into non-overlapping blocks of 4×4 pixels and generate a mapping array containing the locations of the blocks that will embed the feature information of their corresponding blocks by applying the torus automorphism. All the images

This research is supported by National Science Council of ROC under Grant NSC-92-2213-E-126-009.
are assumed to be of size \( N \times N \) pixels and 256 gray-level.

Each block of 4×4 pixels, denoted as \( B \), is further divided into four sub-blocks of 2×2 pixels, as shown in Figure 1(a). The watermark of each sub-block \( B_s \) is a 3-tuple \((v, p, r)\), where \( v \) and \( p \) both are 1-bit authentication watermark for \( B_s \) and \( r \) is a 6-bit recovery watermark for the corresponding sub-block of another block. The following algorithm describes how this 3-tuple watermark is generated and embedded for each sub-block.

### 2.1.1. Sub-Block watermark generation and embedding algorithm

#### 2.1.1.1. Generate the authentication watermark \( v \)

\[
v = \begin{cases} 
1, & \text{if } \text{avg}_{B_s} > \text{avg}_{B} \\
0, & \text{otherwise}
\end{cases}
\]  

where \( \text{avg}_{B_s} \) is the average intensity of \( B_s \) and \( \text{avg}_{B} \) is the average intensity of block \( B \).

#### 2.1.1.2. Generate the parity-check bit \( p \)

\[
p = \begin{cases} 
1, & \text{if num is odd} \\
0, & \text{otherwise}
\end{cases}
\]

where \( \text{num} \) is the total number of 1’s in the six MSB’s of \( \text{avg}_{B_s} \).

From the mapped array generated in preparation step, obtain the location of the block whose recovery information will be stored in block \( B \), and call this block \( A \). Obtain the recovery intensity \( r \) of each corresponding sub-block \( A_s \) within \( A \), respectively, as shown in Figure 1(b). Compute the average intensity of \( A \) and truncating the last two LSB’s from it, as shown in Figure 1(c).

Embed the 3-tuple-watermark \((v, p, r)\), eight bits in all, onto the two LSB’s of each pixel within \( B \), respectively, according to the order as shown in Figure 2.

### 2.2 Hierarchical Tamper Detection

The image is first divided into non-overlapping blocks of 4×4 pixels, as in the watermark embedding process. For each block, we compute the average intensity of the block, denoted as \( \text{avg}_{B_e} \), and truncate the two LSB’s from \( \text{avg}_{B_e} \). We then perform 3-level hierarchical detection as follows.

#### 2.2.1. Hierarchical tamper detection algorithm

**Level-1 detection.** For each 2×2 sub-block within the block \( B_e \), perform the following. 1. Extract its \( v \) and \( p \) stored in the bit 7 of pixel 1,2, respectively. 2. Compute the average intensity of each 2×2 sub-block and truncate the two LSB’s from the computed intensity and denote it as \( \text{avg}_{B_e} \). 3. Count the total number of 1’s in \( \text{avg}_{B_e} \), denoted as \( N_s \). 4. Set the parity-check bit \( \text{p'} \) of sub-block to 1 if \( N_s \) is odd; otherwise, set it to 0. 5. Compare \( p' \) with \( p \). If they are not equal, mark the sub-block as erroneous and complete the detection for this sub-block. 6. Set the algebraic relation \( v' = 1 \) if \( \text{avg}_{B_e} > \text{avg}_{B} \); otherwise, set \( v' = 0 \). 7. Compare \( v' \) with \( v \). If they are not equal, mark the sub-block as erroneous and complete the detection for this sub-block; otherwise, mark it as valid.

**Level-2 detection.** For each block of 4×4, mark the block as erroneous if any of its sub-block is marked erroneous; otherwise, mark it as valid.

**Level-3 detection.** For each valid block of 4×4, mark the block as erroneous if there are five or more erroneous blocks in its 3×3 block-neighborhood, as shown in Figure 3.

---

**Figure 1.** Watermark generation of sub-block comprises of pixel 1,2,3,4.

<table>
<thead>
<tr>
<th>bit 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2.** Embedding the 8-bit Watermark \((v, p, r)\) of pixel 1,2,3,4.

**Level-2 detection.** For each block of 4×4, mark the block as erroneous if any of its sub-block is marked erroneous; otherwise, mark it as valid.

**Level-3 detection.** For each valid block of 4×4, mark the block as erroneous if there are five or more erroneous blocks in its 3×3 block-neighborhood, as shown in Figure 3.

**Figure 3.** 3×3 block-neighborhood of block B

---

**Figure 1.** Watermark generation of sub-block comprises of pixel 1,2,3,4.

<table>
<thead>
<tr>
<th>pixel 1</th>
<th>pixel 2</th>
<th>pixel 3</th>
<th>pixel 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>r</td>
<td>p</td>
<td>r</td>
</tr>
<tr>
<td>v</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>

**Figure 2.** Embedding the 8-bit Watermark \((v, p, r)\) of pixel 1,2,3,4.
(D) Level-4 detection (only required for resisting against VQ attack).

For each valid block $B$ of $4 \times 4$, perform the followings. Calculate the location of the mapping block $C$ using the stored secret key $K$ by torus automorphism. If block $C$ is marked as erroneous, assume $B$ is correct and complete the test for this block.

If block $C$ is valid, perform the followings. For each $2 \times 2$ sub-block within the block $C$, obtain the 6-bit-intensity of each sub-block within block $B$ by extracting from the two LSB’s of the $2 \times 2$ pixels in the corresponding sub-block within block $C$, as illustrated in Figure 2. Denote it as $\text{avg } B_{1}’$. For each $2 \times 2$ sub-block within $B$, compute the average intensity and truncate the two LSB’s from the computed intensity and denote it as $\text{avg } B’$. Compare $\text{avg } B’$ with $\text{avg } B_{1}’$, mark $B$ as erroneous if the two values are different.

2.3. Block-Based tampered image recovery

After the detection stage, all the blocks are marked either valid or erroneous. We’ll only need to recover the erroneous blocks and leave those valid blocks as they are. The restoration for each erroneous block is described in the following algorithm.

Because the information of one block is stored in different block with its location calculated from Torus automorphism, we must find out the stored location for each block that needs recovery. Denote the erroneous block under recovery as block $B$ and the block embedded with its intensity as block $A$. For each invalid block, perform the steps below.

2.3.1. Block recovery algorithm. Calculate the block location of block $A$ using the stored secret key $K$ by torus automorphism. If block $A$ is marked as erroneous, skip the recovery for this block. If block $A$ is valid, obtain the 6-bit-intensity of each sub-block within block $B$ by extracting from the two LSB’s of the $2 \times 2$ pixels in the corresponding sub-block within block $A$. At last step, Pad the 6-bit-intensity with two 0’s and replace each pixel within the sub-block with this new 8-bit intensity. Repeat last step for all four sub-blocks within block $B$ then mark block $B$ as valid.

3. Experiments and Results

3.1. Performance on images with slight to median degree of tampering

The image Beach, as shown in Figure 4(a) has been inserted with a deer, as shown in Figure 4(b). The modification is so natural that one will hardly suspect any changes had been made to this picture. Using our method, we can detect the modification, as shown in (c), (e), and (g) with only level-1, both level-1 and -2, or all three levels of detection, respectively. Figures 4 (d), (f), (h) show the recovered Beaches, respectively.

![Figure 4](image-url)

Figure 4. (a) The watermarked Beach, (b) The tampered Beach, (c)(e)(g) The detected erroneous region, (d)(f)(h) The recovered Beach

3.2. Performance of Tamper Detection on 100% tampered images

We test the tamper detection capability of our scheme with four images, on which all blocks are 100% altered by four alteration methods.

The test results are listed in Table 1 and can be summarized as follows: (a). With only hierarchical level-1 detection, the maximum rate of miss detection is less than 12%. (b). With both level-1 and level-2 detection, the maximum miss detection rate drops significantly to 0.37%. (c). With all level-1, -2, -3 detection, no single tampered block is miss detected.
Table 1. The miss detection rate after each level of inspection

<table>
<thead>
<tr>
<th>Image</th>
<th>Home</th>
<th>Car</th>
<th>Fingerprint</th>
<th>Beach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level-1</td>
<td>11.62%</td>
<td>10.58%</td>
<td>11.84%</td>
<td>11.56%</td>
</tr>
<tr>
<td>Level-2</td>
<td>0.35%</td>
<td>0.37%</td>
<td>0.32%</td>
<td>0.34%</td>
</tr>
<tr>
<td>Level-3</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

3.3. Performance of restoration on distribution of tampered blocks

In this experiment, we want to find out how well one tampered image can be recovered with respect to different sizes and distributions of tampering. The tampered blocks can either be in a form of single tampered chunk, or be in a form of spread tampered blocks. We altered image Lena 10-50% totally with both types of tampering distribution.

The recovering rates of both types of tampering distribution with respect to the rate of tampering are listed in Table 2

Table 2. The recovering rates from both types of tampering

<table>
<thead>
<tr>
<th>Tampering rate</th>
<th>Single-tampered chunk</th>
<th>Spread-tampered blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>95%</td>
<td>99%</td>
</tr>
<tr>
<td>40%</td>
<td>94%</td>
<td>96%</td>
</tr>
<tr>
<td>50%</td>
<td>93%</td>
<td>94%</td>
</tr>
</tbody>
</table>

4. Conclusion

In this paper, we have presented an efficient and effective scheme for both image tamper detection and restoration. Our scheme has three advantages. The first one is that it only uses simple operations such as parity check and comparisons, and thus our method can be executed efficiently. The second advantage is that our method uses a hierarchical-structured detection with the size of the field of view increases accordingly such that the accuracy of tamper localization can be ensured. The third advantage is that our method is very storage effective, as it only requires a secret key $K$ and a public chaotic mixing algorithm to recover a tampered image. The experimental results demonstrate the precision of tamper detection and localization is 99.6% and 100% after level-2 and level-3 inspection. The tamper recovery rate is better than 93% for a 50% or less tampered image. Compare with the method in [8], our method is as simple and as effective in tamper detection and localization, it also provides with the capability of tamper recovery. We hope that we can enhance the security of the scheme significantly in the future.

References