# A New Blind Image Authentication for Image Tampering Detection and Recovery Based on Block-wise Feature Classification

Yung-Chen Chou[1], Chun-Hsiu Yeh[2,3], Jau-Ji Shen[4,5], Jinn-Ke Jan[2]

[1] Dept. of Computer Science and Engineering, Asia University, Taiwan
[2] Dept. of Computer Science and Engineering, National Chung Hsing University, Taiwan
[3] Dept. of Management Information Systems, Chung Chou University of Science and Technology, Taiwan
[4] Dept. of Management Information Systems, National Chung Hsing University, Taiwan
[5] Dept. of Information Communication, Asia University, Taiwan
{yungchen, sheery.yeh}@gmail.com, jjshen@dragon.nchu.edu.tw, jkjan@cs.nchu.edu.tw

## Abstract

This paper presents a blind image authentication technique. It reports on several processes involved in image authentication, tamper detection, and tampered image recovery. The authentication codes of each block consist of three features, variance, grayscale, and edge. To enhance the robustness of each block's authentication codes, these codes will make three copies and permute their data stream before embedding. The proposed blind image authentication method can verify the integrity of an image without requiring access to the original image. Moreover, image recovery for tampered blocks is also conducted with the same authentication code's blocks. The experimental results show that the proposed method can successfully achieve the goal of image authentication and maintain the high visual quality on both of authenticated image and its recovery version if any tamper is happened.

**Keywords:** Blind watermarking, Edge detection, Image authentication, Steganography, Tamper detection and recovery

## 1 Introduction

Data hiding is a commonly used technique for image copyright protection and image authentication issues and which can be briefly classified into three types namely robust watermarking, semi-fragile watermarking, and fragile watermarking. Robust watermarking is typically used to protect intelligent copyrights [1-2]. The main property of robust watermarking is the watermark which embedded into the image can be survived after normal image processing (e.g., JPEG compression, image resizing).

For image authentication purpose, the semi-fragile technique can detect the tampered area of image [3]. Many image authentication methods have been presented in the recent years [3-11]. Nowadays, the image authentication method is not only to deal with the tampered area detection but also its recovery. The main property of fragile watermarking is the embedded watermark sensitivity because it should be damaged when the image is tampered [12-16].

Technically, fragile watermarking can be classified into block-wise [15-16] and pixel-wise [12-13, 17]. Lee and Lin [15] proposed a block-wise watermarking method to embed two watermark copies into an image. When one copy is destroyed, another watermark will be used for tampered area detection. In [16], the most recent successful research results presented an image authentication method which takes the five most significant bits (MSBs) of pixels in a block and cooperating with DCT (discrete cosine transformation) to generate two authentication bits and ten image recovery bits. Additionally, Yang and Shen [18] presented an image authentication and recovery method to preserve the visual quality of the authenticated image. In [18], a VQ (vector quantization) index table of the target image is constructed then the index table is stored for the future tamper detection and tampered area recovery. Lin and Chang [2] presented a frequency domain image authentication method which generated the image feature from the DCT coefficients quantized by JPEG quantization table.

There are two ways to use authentication codes to perform digital image authentication: the first way is to send the authentication code and the image separately to the receiver and then compare them to see if the image has been tampered with [4]. The second way is to embed the authentication code within the image itself and then extract the authentication code from the image to determine whether or not the image has been tampered with [12-13, 16-17, 20]. The watermark is

*Corresponding Author: Chun-Hsiu Yeh; E-mail: sheery.yeh@gmail.com
DOI: 10.3966/160792642018111906026

calculated by original image itself that is call self-embedding [12, 16, 20, 24].

In this paper, a blind image authentication scheme is proposed to achieve the goal of image integrity verification by tamper detection and tampered area recovery. The proposed method generates image feature information by image block variance, grayscale, and edge direction information. Blind image authentication is defined as to verify the image integrity without using original image [19]. The remainder parts of this paper are organized as follows. Some background knowledge related the proposed method is described in Section 2. Section 3 shows the proposed method in detail. After that, Section 4 summarizes the experimental results. Finally, we make some conclusions in Section 5.

## 2 Related Works

### 2.1 Side-Match Vector Quantizatio

SMVQ (Side-match vector quantization) [25] scheme uses the margin pixels of two neighbor blocks (i.e., upper side and left side) to predict a suitable codeword. For example, the block X is unknown then *X1*, *X2*, *X3*, *X4*, and *X7* are the margin pixels in the block X. So, *X1* = (*U7* + *L3*) div 2, *X2* = *U8*, *X3* = *U9*, *X4* = *L6*, and *X7* = *L9* (as Figure 1). After that, SMVQ uses the predict pixel values to construct a state codebook and choose a codeword most similar with the predict pixels.

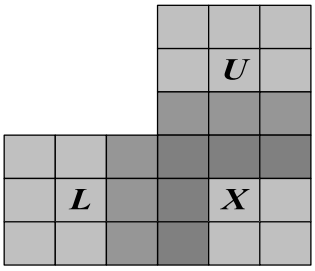



**Figure 1.** The concept diagram of SMVQ

### 2.2 Edge Detection

Edge detection is also an important operation in our research. It primarily uses an image grayscale to determine whether an edge exists, since edge pixels have a higher grayscale. Sobel edge detection filters out relatively unimportant data from the image [26], such that the edges in the image remain (Figure 2). In this paper, four basic edge detection masks, horizontal, vertical, 45° and 135° of Sobel filter are adapted to generate feature code of each image block. The size of each mask can be set as 3 × 3 and *z1* to *z9* are the pixel values of the block (Figure 3(a)). In order to detect the different direction edges, the different weighted pattern of each edge mask is given as shown Figure 3(b) to Figure 3(f). *Gx* is used to detect the grayscale in the horizontal direction, as in Equation (1). *Gy* is used to detect the grayscale in the vertical direction, as in Equation (2). The grayscale magnitude $\nabla f$ is calculated using *Gx*, *Gy*, and the different weights given to pixel values, as in Equation (3). If $\nabla f$ is greater than the threshold value, then the block has an edge in it.

(a) Original lena image (b) Lena image after sobel detection

**Figure 2.** Sobel detection of Lena image

| | | |
|---|---|---|
| *z1* | *z2* | *z3* |
| *z4* | *z5* | *z6* |
| *z7* | *z8* | *z9* |

(a) Pixels positions of block

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

(b) Horizontal mask

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

(c) Vertical mask

| | | |
|---|---|---|
| 0 | -1 | -2 |
| 1 | 0 | -1 |
| 2 | 1 | 0 |

(d) 45 degree mask

| | | |
|---|---|---|
| 2 | 1 | 0 |
| 1 | 0 | -1 |
| 0 | -1 | -2 |

(e) 135 degree mask

| | | |
|---|---|---|
| -1 | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | -1 |

(f) Non-edge mask

**Figure 3.** The masks of sobel edge detection

$$Gx = (z1 + 2 \times z2 + z3) - (z7 + 2 \times z8 + z9) \quad \textbf{(1)}$$

$$Gy = (z3 + 2 \times z6 + z9) - (z1 + 2 \times z4 + z7) \quad \textbf{(2)}$$

$$\nabla f = \sqrt{G_x^2 + G_y^2} \quad \textbf{(3)}$$

## 3 Proposed Method

This paper presents a novel image authentication method that uses three significant features as criteria for maintaining image integrity. There are two phases in our scheme: (1) authentication code generating and

embedding, (2) tamper detection and recovery process. The terms used in our proposed method are defined in Table 1.

**Table 1.** Summary of terms

| Notation | Description |
|---|---|
| $O$ | Original image |
| $b_i$ | $i$-th block of original image |
| $Vb_i$ | The variance value of block $b_i$ |
| $Gb_i$ | The grayscale value of block $b_i$ |
| $Eb_i$ | The chosen edge direction of block $b_i$ |
| $FVb_i$ | Variance feature code of $b_i$ |
| $FGb_i$ | Grayscale feature code of block $b_i$ |
| $FEb_i$ | Edge feature code of block $b_i$ |
| $q1$ | First grayscale variance value $b_i$ |
| $q2$ | Second grayscale variance value of block $b_i$ |
| $q3$ | Third grayscale variance value of block $b_i$ |
| $F$ | The original image's feature code |
| $F'$ | The original image's feature code after permutation |
| $FA$ | The feature code generated from authenticated image |
| $E_v$ | Edge value generated by applying vertical edge mask |
| $E_h$ | Edge value generated by applying horizontal edge mask |
| $E_{45}$ | Edge value generated by applying 45-degree edge mask |
| $E_{135}$ | Edge value generated by applying 135-degree edge mask |
| $p$ | The number of blocks of host image |

## 3.1 Authentication Code Generating and Embedding

The proposed method presents novel block-wise feature code for image authentication. The feature code of a block consists of the variance, grayscale, and edge , the concept of authentication code generation show in Figure 4 and Algorithm 1.

### 3.1.1 Variance Feature Code

First, the original image $O$ is divided into non-overlapping blocks of size $n \times n$ and denoted as $O = \{b_i \mid i = 1, 2, \ldots, p\}$, where the variance feature $Vb_i$ of each $b_i$ is calculated.

The content of a natural image can be classified as complex or smooth. The variance of each block (see Equation (4)) provides guidance for block type classification. In Equation (4), $x_j$ is a pixel in block $b_i$. Because the variance range of an image block is very wide, so it is divided into four sub-ranges, and each sub-range is represented by two bits. The quartile concept is used as the variance sub-ranges setting rule. According to our pretests, the best choices of three mark points of quartile are located at 25% ($q1$), 50% ($q2$) and 80%($q3$) which are capable to classify image blocks properly via their texture complexity (as step 2 of Algorithm 1).

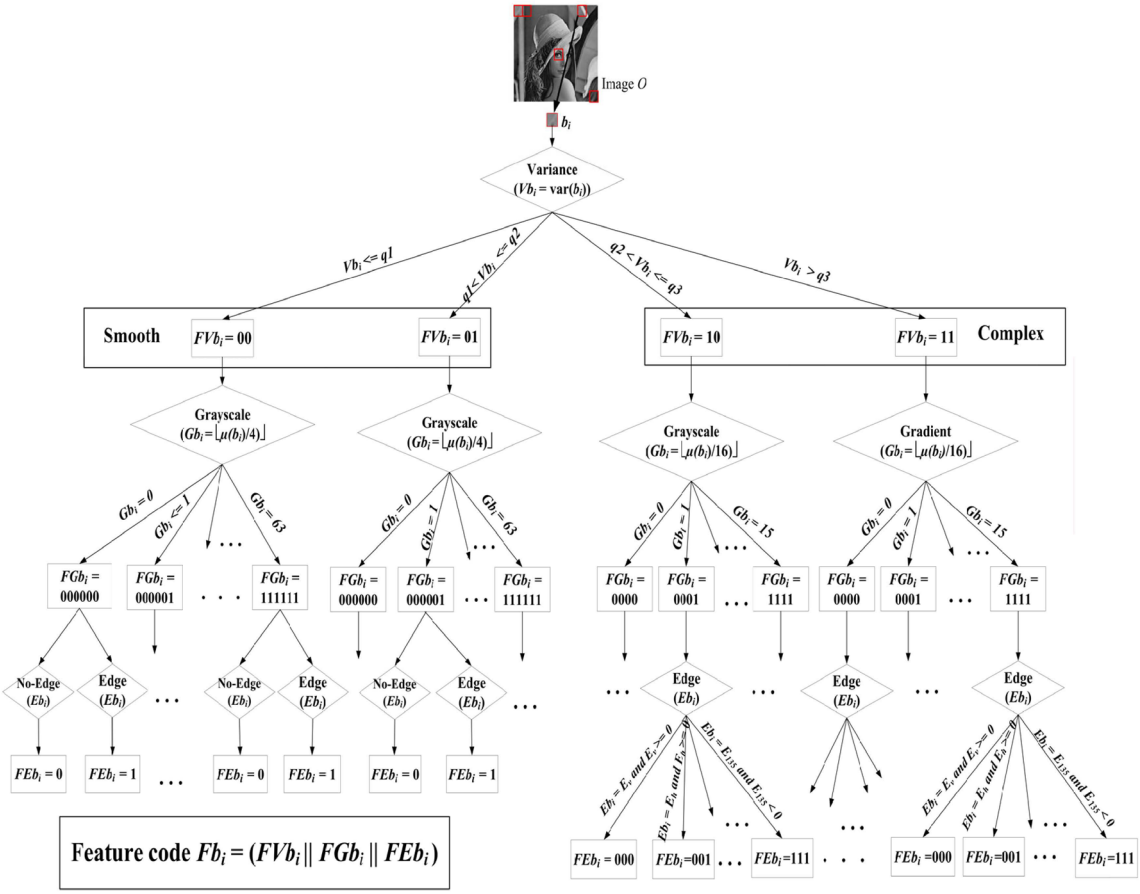



**Figure 4.** The concept diagram for the proposed feature code generating process

**Algorithm 1.** Authentication Codes Generating and Embedding Procedure

Input: Original image $O$
Output: Authenticated image $O'$
Step 1: Image $O$ is firstly divided into non-overlapping blocks with n x n sizes. Let $b_1, b_2, ..., b_p$ be the blocks;
Step 2: Calculate the variance $Vb_i$ of each block $b_i$ and classify $b_i$ as smooth or complex block when $Vb_i$ is located at the range before or after $q2$;
Step 2.1: For $i$ =1 to $p$
Step 2.2: If $Vb_i \le q1$ then $FVb_i$ = '00'; // generating variance feature code
Elseif $q1 < Vb_i \le q2$ then $FVb_i$ = '01';
Elseif $q2 < Vb_i \le q3$ then $FVb_i$ = '10';
Else $q3 < Vb_i$ then $FVb_i$ = '11';
End If;
Step 3: Calculate the grayscale $Gbi$ of each block $b_i$ by taking mean values of all pixels in $b_i$. Use four bits or six bits to encode the grayscale value as which is generated from the complex or smooth blocks by Equation (6) or (7), respectively;
Step 3.1: If $FVb_i$ = '10' or '11' then $Gb_i$ = floor($\mu$ ($b_i$) / 16) // generating grayscale feature code for complex block
Step 3.2: If $Gb_i$ = 0 then $FGb_i$ = '0000';
Elseif $Gb_i$ = 1 then $FGb_i$ = '0001';
**...** ;
Else $Gb_i$ = 63 then $FGb_i$ = '1111';
Endif ;
Endif;
Step 3.3: If $FVb_i$ = '00' or '01' then $Gb_i$ = floor($\mu(b_i)$ / 4) // generating grayscale feature code for smooth block
Step 3.4: If $Gb_i$ = 0 then $FGb_i$ = '000000';
Elseif $Gb_i$ = 1 then $FGb_i$ = '000001';
**...** ;
Else $Gb_i$ = 15 then $FGb_i$ = '111111';
Endif ;
Endif;
Step 4: Generate edge feature code $FEb_i$ using Equation (8) and (9). Three bits are used to represent the edge direction in each block of complex regions, and one bit is used to indicate whether this block has edge in smooth region; End For;
Step 5: Generate authentication code $F= FVb_i \| FGb_i \| FEb_i$, where i = 1, 2, …, p, '||' means concatenated together with three feature codes;
Step 6: $F'$ = Permute($q1 \| q2 \| q3 \| F \| F \| F$, *key*), where 'key' denotes a selected seed for the random permutation function;
Step 7: Embed $F'$ into each $O$'s pixel to create the authenticated image $O'$ by any suitable data hiding scheme.

$$Vb_i = \frac{\sum_{j=0}^{n\times n-1}(x_j - \mu)}{n \times n} \quad \textbf{(4)}$$

$$\mu = \frac{\sum_{j=0}^{n\times n-1} x_j}{n \times n} \quad \textbf{(5)}$$

### 3.1.2 Grayscale Feature Code

The grayscale feature is the mean pixel value of a block (see Equation (5)). We aim to make this feature code sensitive to block content modification. Thus, the grayscale feature generated for the complex block is represented by four bits and for a smooth block is represened by six bits. For complex content block, its $\mu$ is encoded by dividing the gray scale value range (i.e., 0~255) into $2^4$ segments to get $\mu$'s located segment (refer to Equation (6)) to be the grayscale feature. For smooth content block, its $\mu$ is encoded by dividing the gray scale value range (i.e., 0~255) into $2^6$ segments to get $\mu$'s located segment (refer to Equation (7)) to be the grayscale feature (as step 3 of Algorithm 1).

$$\text{Complex blocks: } FGb_i = [(\frac{\mu}{16})_2] \quad \textbf{(6)}$$

$$\text{Smooth blocks: } FGb_i = [(\frac{\mu}{4})_2] \quad \textbf{(7)}$$

### 3.1.3 Edge Feature Code

The third part of the block feature is denoted by edge directions. The proposed method use Sobel detection to generate the edge feature. We use four bits and one bit to represent the edge feature of complex content block and smooth content block, respectively. The vertical, horizontal, 45 degree, and 135 degree masks (refer to Figure 2) are used to determinate the edge direction in case of $b_i$ as a complex content block. The non-edge mask (Figure 3(f)) is used to generate the edge feature of a smooth block. Moreover, different grayscale pixel values distribution might lead the same edge direction, such as a block has dark and bright at two either sides. That is the reason why one more bit is required to indicate the edge direction of a complex

content block. Equation (8) shows the edge direction choosing policy of the proposed edge feature generating, and Equation (9) shows the rule in generating the edge feature code (as step 4 of Algorithm 1).

$$Eb_i = \max\{|E_v|, |E_h|, |E_{45}|, |E_{135}|\} \quad \textbf{(8)}$$

$$FEb_i = \begin{cases} 000_2, if\ Eb_i = E_v \ \ and\ \ E_v \geq 0\ and\ Vb_i > q2 \\ 001_2, if\ Eb_i = E_h \ \ and\ \ E_h \geq 0\ and\ Vb_i > q2 \\ 010_2, if\ Eb_i = E_{45}\ and\ \ E_{45} \geq 0\ and\ Vb_i > q2 \\ 011_2, if\ Eb_i = E_{135}\ and\ E_{135} \geq 0\ and\ Vb_i > q2 \\ 100_2, if\ Eb_i = E_v \ \ and\ E_v \ \ < 0\ and\ Vb_i > q2 \\ 101_2, if\ Eb_i = E_h \ \ and\ E_h \ \ < 0\ and\ Vb_i > q2 \\ 110_2, if\ Eb_i = E_{45}\ and\ E_{45}\ < 0\ and\ Vb_i > q2 \\ 111_2, if\ Eb_i = E_{135}\ and\ E_{135} < 0\ and\ Vb_i > q2 \\ 0, \quad if\ |Eb_i| > 0\ and\ Vb_i \leq q2 \\ 1, \quad if\ Eb_i = 0\ and\ Vb_i > q2 \end{cases} \quad \textbf{(9)}$$

Finally, each block of the authenticating image has been generated its variance, grayscale, and edge feature codes, denoted as $FVb_i$, $FGb_i$, and $FEb_i$ three feature codes, respectively. For each block, all of its feature codes will be concatenated together as its final authentication code (shown as step 5 of Algorithm 1).

#### 3.1.4 Authentication Code Embedding Process

In order to enhance the robustness of image authentication codes, the feature code is duplicated three times and permuting by a selected key (as step 6 of Algorithm 1) before applying data embedding method to conceal the authentication codes into host image. Note that, any data hiding method can be used in the proposed method at this phase, such as LSB [17, 21, 23], DCT [20], and modulus function [22]. An image is said to be authenticated when its authentication codes have been embedded into all of its pixels.

### 3.2 Tamper Detection and Recovery Process

The scheme of tampered blocks detection and its recovery is consisted in four processes. First, the embedded authentication code shall be extracted back from $O'$ according to the same information hiding scheme applied at Algorithm 1 step 7. Second, a new feature data of $O'$ will be regenerated by the proposed authentication code generating procedure (as step 1 to step 4 of Algorithm 1). Third, a comparison will be made between extracted and new generated authentication codes to determine whether the block has been tampered or not. Finally, for each tampered block recovery, the SMVQ concept (mentioned in Figure 1) will be applied to find the best fit block from all untampered blocks, which have the same extracted authentication code with this tampered block, such that the selected makeup block is the most connected with its untampered neighbors on $O'$. This SMVQ process will be taken several phases to recover $O'$ until no tampered block can be recovered by this process.

**Algorithm 2.** Image Tamper Detection and Recovery Procedure

Input: Authenticated image $O'$
Output: Recovered image $O''$
Step 1: Extract the permuted authentication code $F'$ from $O'$ using the same data hiding scheme in Algorithm 1 step 7;
Step 2: Permute $F'$ back as the normal version of authentication code, said $F''$, by the same key selected in Algorithm 1 step 6;
Step 3: Sepatate $F''$ into three parts $F1'$, $F2'$, and $F3'$ with respect to the three copies of $F$;
Step 4: Regenerate feature code $FA$ from $O'$ with the same steps from step1 to step 5 of Algorithm 1;
Step 5: Let $N_F$ is the total bits of $F1'$;
Step 5.1: Let $FR$ be the authentication code recomposed from $F1'$, $F2'$, and $F3'$;
Step 5.2: For j = 0 to ($N_F$ – 1)
Step 5.3: if ($F1_j' + F2_j' + F3_j'$) > 1 then $FR_j$ = 1; // using voting strategy check the accuracy of the feature code bits.
else $FR_j$ = 0;
End if;
End for;
Step 6: Detect the tampered blocks by comparing the difference between $FA$ and $FR$;
Step 6.1: For i = 1 to p
Step 6.2: Let $FAb_i'$ be the nine bits authentication code of $b_i$ found from $FA$; Let $FRb_i'$ be the nine bits authentication code generated from voting result of $FR$;
Step 6.3: if $FAb'_i \neq FRb_i'$ then label this block $b_i$ as a tampered block;
End for
Step 7: For i = 1 to p
Step 7.1: if $b_i$ is a tampered block then recover $b_i$ by the SMVQ concept mentioned in the first paragraph of Subsection 3.2;
End for;
Step 8: If there any tampered block is not restored, then this block is restored by cubic interpolation method proposed in [6];
Step 9: Output recovered image $O''$.

Although, most of the tampered blocks can be recovered by using the previous SMVQ process. However, in some cases, there are some tampered blocks without any other same authentication code block can be restored, the cubic interpolation method proposed in [6] will be adapted to handle this case.

## 4 Experimental Results

To evaluate the performance of our proposed method, the algorithms were implemented by Octave software on an Ubuntu 16.04 system. Eight commonly used test gray images sized 512 × 512 were used in our simulations (Figure 5). The test images included smooth (e.g., Toys) and complex content (e.g., Baboon). Consider the convenience of the experiment, we use the LSB replacement scheme which is a simple commonly used data embedding technique [17, 21]. The permuted authentication code is embedded into three LSBs of every pixel by direct value substitution. Of course, if we use better information hiding technology, the experiemental results will be better. Also, three and two LSB bits were set to embed the image feature data for block size 3 × 3 and 5 × 5, respectively.

(a) Baboon (b) Barbara (c) Boat (d) Cameraman

(e) Jet(F16) (f) Lena (g) Toys (h) Zelda

**Figure 5.** Eight test images

The original image is called an authenticated image when it is embedded its own authentication code. The image in which the secret message is embedded is known as the cover or host image, and the secret message or information is known as the watermark [16]. The quality standard of an image was evaluated using PSNR (peak signal-to-noise ratio) in Equation (10). Higher values of PSNR represent better image quality. Structural similarity (SSIM) is used for measuring the similarity between two images in Equation (12).

$$\mathrm{PSNR} = 10 \times \log \frac{255^2}{\mathrm{MSE}} \qquad \textbf{(10)}$$

$$\mathrm{MSE} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} (O_{i,j} - Y_{i,j})^2 , \qquad \textbf{(11)}$$

where $H$ x $W$ is the size of the image, $O_{p,q}$ is the pixel value refers to the original image position (p, q), $Y_{p,q}$ is the pixel value refers to the recovery image position.

$$\mathrm{SSIM} = \frac{(2\mu_O\mu_Y + c_1)(2\sigma_{OY} + c_2)}{(\mu_O^2 + \mu_Y^2 + c_1)(\sigma_O^2 + \sigma_Y^2 + c_2)} \qquad \textbf{(12)}$$

$\mu_O$ and $\mu_Y$ are the principal values of the original host and the compared images, respectively. Parameters $\sigma_O^2$ and $\sigma_Y^2$ are the variances of the original host image $O$ and the compared image $Y$, respectively. The parameter $\sigma_{OY}$ is the covariance of the original image $O$ and the compared image $Y$. Variables $c_1$ and $c_2$ are for stabilizing the division with weak denominators computed. $L$ is the dynamic range of the pixel values. A SSIM value close to one indicates the recovered image is very similar to the original image.

For tamper detection, a false positive (FP) error is a result that indicates a given condition has been fulfilled when it has not been fulfilled. A false negative (FN) error, is a result that indicates that a condition failed when it was successful (i.e., erroneously no effect was assumed). Here, an FP involves counting the number of tampered blocks that were not recovered by the proposed recovery procedure. Additionally, an FN involves counting the number of correct blocks indicated as tampered.

In the cases of image tampering by adding an attached butterfly image or text NCHU into authenticated images of Barbara and Zelda (Figure 6(b) and Figure 7(b)), the experimental results show that our method not only accurately examines tampered regions in relation to an authenticated image (Figure 6(c) and Figure 7(c)), but also effectively conducts recovery in tampered regions (Figure 6(d) and Figure 7(d)). In Table 2, the experiment shows the results about a butterfly image and text NCHU being attached into the Barbara, Zelda and Lena images. There are some FP blocks because of their non-regular block's shape or with the same color as original image. The NTB means number of tampering blocks, and the TDAR means tampering detection accuracy rate in Equation (13) .

(a) Authenticated image (b) Attack Image butterfly (c) Tamper detected (d) Recovered image

**Figure 6.** Results of tampering detection and recovery of the barbara image


(a)Authenticated image (b) Attack text NCHU (c) Tamper detected (d) Recovered image

**Figure 7.** Results of tamper detection and recovered zelda image

$$\text{TDAR} = \frac{(\text{Number of tampering block} - \text{FP})}{\text{Number of tamper blocks}} \quad \textbf{(13)}$$

We conducted the experiments by cropping the attack pixel value of the authenticated image at 5% – 30%. for instance, cropping pixels pixels in the Lena image to reach 20%, with tampering about 5,849 blocks. Our proposed method for tamper detection has an accuracy rate of 100%; the PSNR of recovered image is 41.69 dB corresponding to the authenticated image, and its SSIM is 0.944 with the host image, the NTB is number of tampered blocks, as shown in Table 3. Figure 8(a) to Figure 8(f) show different rates of cropping pixels on Lena image. The results of the image tampering detection are shown in Figure 9. The results of the image recovery are shown in Figure 10. Additionally, we set the block size to 5 × 5, using eight authenticated images to detect tampering by comparing PSNR and SSIM, as shown in Table 4. The experimental results in Table 3 and Table 4 show that the complex (smooth) image has the better recovery image quality than the smooth (complex) one by authenticating with a more larger (smaller) block size. We want to emphasize that if an information hiding technology capable to retain a high quality cover image is applied in our proposed scheme then a more higher quality recovery image can be expected.

**Table 2.** Experimental results of image recovery after tampering with different attack images

| Attack image | Images | PSNR of authentication images | Number of tampered blocks | Performance of tampered recovery image | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | PF | FN | TDAR (%) | PSNR (dB) | SSIM |
| butterfly | Lena | 41.72 | 887 | 1 | 0 | 99.9 | 52.47 | 0.993 |
| | Barbara | 41.45 | 888 | 7 | 0 | 99.2 | 50.47 | 0.994 |
| | Zelda | 41.14 | 888 | 6 | 0 | 99.3 | 50.11 | 0.992 |
| NCHU (text) | Lena | 41.72 | 608 | 10 | 0 | 98.4 | 51.05 | 0.995 |
| | Barbara | 41.45 | 608 | 4 | 0 | 99.3 | 51.99 | 0.996 |
| | Zelda | 41.14 | 604 | 8 | 0 | 98.7 | 51.72 | 0.994 |

**Table 3.** Result of recovered images in the tampered areas with different tampering rates by block size is 3 × 3

| Cover images | PSNR of authenticted. Images (dB) | Performance of recovery images | Tampering rates | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 5% | 10% | 15% | 20% | 25% | 30% |
| Baboon | 41.49 | FP / FN | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 |
| | | NTB | 1530 | 2925 | 4420 | 5849 | 7310 | 8773 |
| | | TDAR(%) | 100 | 100 | 100 | 100 | 100 | 100 |
| | | PSNR (dB) | 41.56 | 39.16 | 37.52 | 36.11 | 35.29 | 34.59 |
| | | SSIM | 0.982 | 0.962 | 0.944 | 0.920 | 0.899 | 0.880 |
| Barbara | 41.45 | FP / FN | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 |
| | | NTB | 1530 | 2925 | 4420 | 5849 | 7310 | 8773 |
| | | TDAR(%) | 100 | 100 | 100 | 100 | 100 | 100 |
| | | PSNR (dB) | 44.98 | 41.71 | 40.06 | 39.25 | 38.48 | 37.80 |
| | | SSIM | 0.990 | 0.980 | 0.968 | 0.958 | 0.946 | 0.935 |
| Boat | 41.58 | FP / FN | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 |
| | | NTB | 1530 | 2925 | 4420 | 5849 | 7310 | 8773 |
| | | TDAR(%) | 100 | 100 | 100 | 100 | 100 | 100 |
| | | PSNR (dB) | 52.40 | 49.17 | 47.14 | 45.16 | 43.51 | 42.25 |
| | | SSIM | 0.993 | 0.984 | 0.973 | 0.962 | 0.953 | 0.941 |
| Cameraman | 40.47 | FP / FN | 0/0 | 0/0 | 0/0 | 14/0 | 34/0 | 52/0 |
| | | NTB | 1530 | 2925 | 4420 | 5791 | 7210 | 8636 |
| | | TDAR(%) | 100 | 100 | 100 | 99.76 | 99.53 | 99.40 |
| | | PSNR (dB) | 46.19 | 43.81 | 41.08 | 41.31 | 39.07 | 38.71 |
| | | SSIM | 0.983 | 0.968 | 0.947 | 0.931 | 0.911 | 0.895 |
| Jet(F16) | 42.30 | FP / FN | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 |
| | | NTB | 1530 | 2925 | 4420 | 5849 | 7310 | 8773 |
| | | TDAR(%) | 100 | 100 | 100 | 100 | 100 | 100 |
| | | PSNR (dB) | 47.35 | 44.98 | 44.22 | 42.36 | 42.01 | 42.03 |
| | | SSIM | 0.987 | 0.976 | 0.964 | 0.949 | 0.940 | 0.926 |

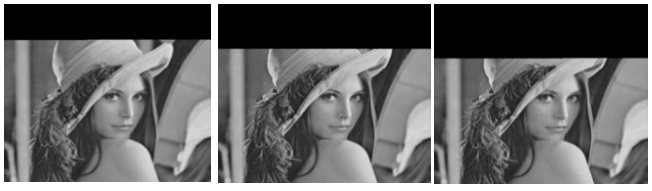
**Table 3.** Result of recovered images in the tampered areas with different tampering rates by block size is 3 × 3 (continue)

| Cover images | PSNR of authenticted. Images (dB) | Performance of recovery images | Tampering rates | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 5% | 10% | 15% | 20% | 25% | 30% |
| Lena | 41.72 | FP / FN | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 |
| | | NTB | 1530 | 2925 | 4420 | 5849 | 7310 | 8773 |
| | | TDAR(%) | 100 | 100 | 100 | 100 | 100 | 100 |
| | | PSNR (dB) | 49.76 | 45.15 | 42.92 | 41.69 | 40.17 | 39.90 |
| | | SSIM | 0.989 | 0.975 | 0.960 | 0.944 | 0.929 | 0.913 |
| Toys | 40.79 | FP / FN | 0/0 | 0/0 | 0/0 | 1/0 | 0/0 | 1/0 |
| | | NTB | 1530 | 2925 | 4420 | 5849 | 7310 | 8773 |
| | | TDAR(%) | 100 | 100 | 100 | 99.99 | 100 | 99.99 |
| | | PSNR (dB) | 48.42 | 47.48 | 43.83 | 42.98 | 42.19 | 40.98 |
| | | SSIM | 0.987 | 0.976 | 0.965 | 0.952 | 0.946 | 0.932 |
| Zelda | 41.14 | FP / FN | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 |
| | | NTB | 1530 | 2925 | 4420 | 5849 | 7310 | 8773 |
| | | TDAR(%) | 100 | 100 | 100 | 100 | 100 | 100 |
| | | PSNR (dB) | 49.23 | 45.84 | 42.89 | 41.44 | 40.47 | 39.58 |
| | | SSIM | 0.990 | 0.978 | 0.963 | 0.948 | 0.934 | 0.924 |

**Table 4.** Result of recovered images in the tampered areas with different tampering rates by block size is 5 × 5

| Cover images | PSNR of auth. images | Performance of recovery images | Tampering rates | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 5% | 10% | 15% | 20% | 25% | 30% |
| Baboon | 50.28 | PSNR (dB) | 40.77 | 38.28 | 36.57 | 35.35 | 34.45 | 33.85 |
| | | SSIM | 0.968 | 0.933 | 0.896 | 0.862 | 0.828 | 0.799 |
| Barbara | 50.23 | PSNR (dB) | 43.70 | 39.95 | 38.76 | 37.74 | 37.11 | 36.50 |
| | | SSIM | 0.984 | 0.962 | 0.948 | 0.930 | 0.916 | 0.897 |
| Boat | 50.12 | PSNR (dB) | 51.56 | 45.38 | 44.93 | 38.39 | 36.80 | 36.39 |
| | | SSIM | 0.996 | 0.986 | 0.981 | 0.927 | 0.908 | 0.894 |
| Cameraman | 49.60 | PSNR (dB) | 46.25 | 43.44 | 42.72 | 40.50 | 38.79 | 38.02 |
| | | SSIM | 0.983 | 0.965 | 0.947 | 0.929 | 0.902 | 0.880 |
| Jet (F16) | 50.84 | PSNR (dB) | 46.42 | 43.14 | 42.28 | 42.14 | 41.43 | 41.02 |
| | | SSIM | 0.987 | 0.974 | 0.964 | 0.959 | 0.947 | 0.936 |
| Lena | 50.42 | PSNR (dB) | 51.06 | 47.52 | 44.23 | 42.22 | 40.90 | 39.81 |
| | | SSIM | 0.995 | 0.987 | 0.972 | 0.959 | 0.943 | 0.925 |
| Toys | 49.60 | PSNR (dB) | 53.01 | 41.88 | 41.09 | 40.81 | 40.31 | 39.50 |
| | | SSIM | 0.993 | 0.951 | 0.940 | 0.935 | 0.929 | 0.918 |
| Zelda | 49.99 | PSNR (dB) | 47.58 | 44.78 | 42.17 | 41.07 | 39.43 | 38.84 |
| | | SSIM | 0.990 | 0.978 | 0.963 | 0.949 | 0.933 | 0.919 |

(a) 5% (b) 10% (c) 15%



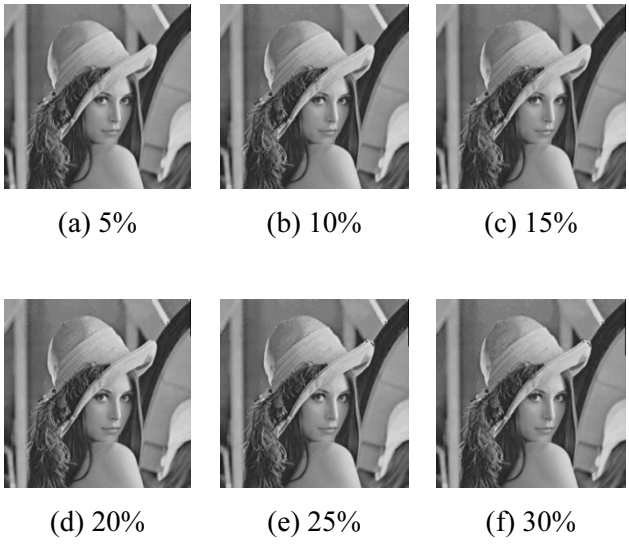
(d) 20% (e) 25% (f) 30%

**Figure 8.** Different tampered rates with lena image

(a) 5% (b) 10% (c) 15%

(d) 20% (e) 25% (f) 30%

**Figure 9.** Tampered detection result with lena image at different tampering rates



(a) 5% (b) 10% (c) 15%

(d) 20% (e) 25% (f) 30%

**Figure 10.** Results of recovered image after tamper detection at different tampering rates

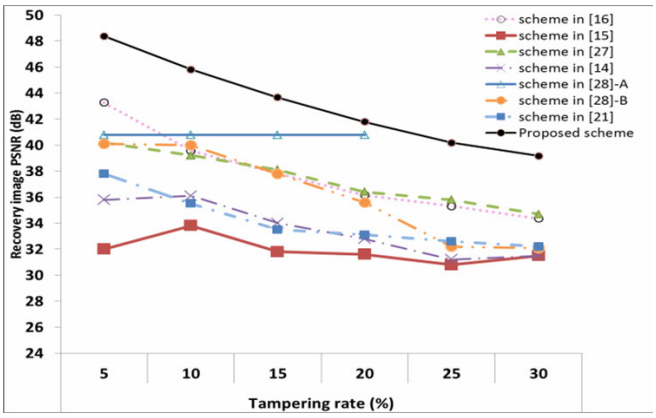



**Figure 11.** Experimental lena images: comparison of image PSNR of the proposed method and different methods with different tampering rates

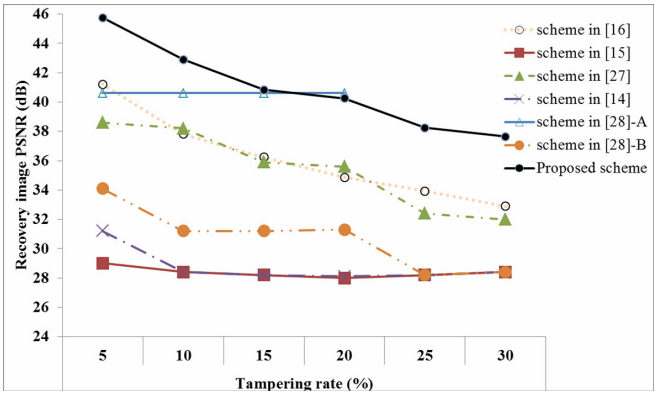



**Figure 12.** Experimental cameraman images: comparison of image PSNR of the proposed method and different methods with different tampering rates

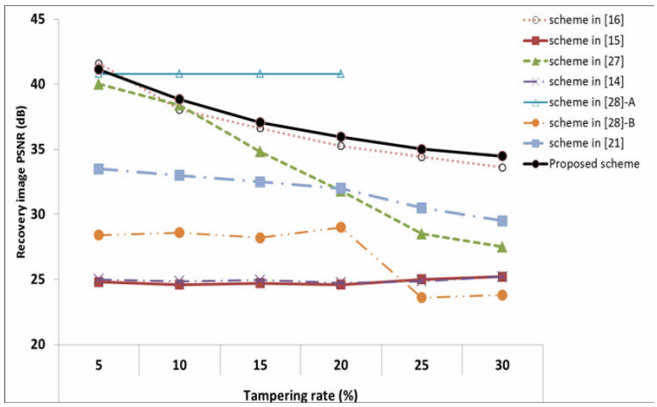



**Figure 13.** Experimental baboon images: comparison of image PSNR of the proposed method and different methods with different tampering rates

Our proposed scheme is also compared to six schemes proposed in [14-16, 21, 27-28]. For the three images, Baboon, Lena, and Cameraman, a size 256 × 256 image was used to compare the PSNR of the recovered images with our proposed methods at different tampering rates from 5%-30%. Those schemes used different block sizes to experiment on, for example, 2 × 2 block is applied in [16], while 8 × 8 block is conduct in [21]. In the scheme of [15], the recovery image uses the mean value of the block for retrieval when the content replacement is not too extensive. In Figure 13, the scheme [28]-A has more better recovery image quality, because they can restore the original 5 MSB data of image for a relatively small tampering rate such as below 24%. Figures 11 to 13 show that our proposed method has more better recovery quality of the tampered image than the six schemes at different tampering rates of 5%-30%.

## 5 Conclusions

Digital images can be delivered easily over the internet to any destination. Thus, determining how to maintain the content integrity of digital image is an important issue. This paper presented a novel image authentication method that uses three significant features as the criteria for maintaining image integrity. The experimental results show that our method can perform the images tamper detection successfully, and the high visual quality is not only maintained for the authenticated image but also for its recovery version if necessary. Most importantly, the proposed method successfully authenticated image content without using the original image's information.

## References

[1] F. Ahmed, M. Y. Siyal, V. U. Abbas, A Secure and Robust Hash-Based Scheme for Image Authentication, *Signal Processing*, Vol. 90, No. 5, pp. 1456-1470, May, 2010.

[2] C. Y. Lin, S. F. Chang, A Robust Image Authentication

Method Distinguishing JPEG Compression from Malicious Manipulation, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 2, pp. 153-168, February, 2001.

[3] X. Zhou, X. Duan, D. Wang, A Semi-Fragile Watermark Scheme for Image Authentication, *Proceedings of IEEE 10th International Multimedia Modeling Conference*, Brisbane, Australia, 2004, pp. 374-377.

[4] L. Dong, X. Kong, B. Wang, X. You, A Robust JPEG Image Tampering Detection Method Using GLCM Features, *Advances in Information Sciences and Service Sciences*, Vol. 3, No. 10, pp. 348-391, November, 2011.

[5] P. W. Wong, N. Memon, Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification, *IEEE Transaction on Image Processing*, Vol. 10, No. 10, pp. 1593-1601, October, 2001.

[6] R. G. Keys, Cubic Convolution Interpolation for Digital Image Processing, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 29, No. 6, pp. 1153-1160, December, 1981.

[7] R. J. Hwang, T. K. Shih, C. H Kao, A Lossy Compression Tolerant Data Hiding Method Based on JPEG and VQ, *Journal of Internet Techology*, Vol. 5, No. 3, pp. 171-178, July, 2004.

[8] J. C. Chuang, Y. C. Hu, C. C. Lo, W. L. Chen, Grayscale Image Tamper Detection and Recovery Based on Vector Quantization, *International Journal of Security and its Applications*, Vol. 7, No. 6, pp. 209-228, November, 2013.

[9] S. Dadkhah, A. A. Manaf, Y. Hori, A. E. Hassanien, S. Sadeghi, An Effective SVD-based Image Tampering Detection and Self-recovery Using Active Watermarking, *Signal Process: Image Communication*, Vol. 29, No.10, pp. 1197-1210, November. 2014.

[10] C. M. Wu, Y. C Hu, K. Y. Liu, J. C. Chuang, A Novel Active Image Authentication Scheme for Block Truncation Coding, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 7, No. 5, pp. 13-26, October, 2014.

[11] W. C. Wu, Z. W. Lin, SVD-Based Self-Embedding Image Authentication Scheme Using Quick Response Code Features, *Journal of Visual Communication and Image Representation*, Vol. 38, No. 7, pp. 18-28, July, 2016.

[12] D. Singh, S. Shivani, S. Agarwal, Self-Embedding Pixel Wise Fragile Watermarking Scheme for Image Authentication, *Intelligent Interactive Technologies and Multimedia,* Vol. 276, pp. 111-122, January, 2013.

[13] Y. Z, He, Z. Han, A Fragile Watermarking Scheme with Pixel-wise Alteration Localization, *2008 9th International Conference on Signal Processing*, Beijing, China, 2008, pp. 2201-2204.

[14] X. Zhang, S. Wang, G. Feng, Fragile Watermarking Scheme with Extensive Content Restoration Capability, *IWDW '09 Proceedings of the 8th International Workshop on Digital Watermarking*, Guildford, England, 2009, pp. 268-278.

[15] T. Y. Lee, S. D. Lin, Dual Watermark for Image Tamper Detection and Recovery, *Pattern Recognition*, Vol. 41, No. 11, pp. 3497-3506, November, 2008.

[16] D. Singh, S. K. Singh, Effective Self-embedding Watermarking Scheme for Image Tampered Detection and Localization with Recovery Capability, *Journal of Visual Communication and Image Representation*, Vol. 38, pp. 775-789, July, 2016.

[17] R. Z. Wang, C. F. Lin, J. C. Lin, Image Hiding by Optimal LSB Substitution and Genetic Algorithm, *Pattern Recognition*, Vol. 34, No. 3, pp. 671-683, March, 2001.

[18] C. W. Yang, J. J. Shen, Recover the Tampered Image Based on VQ Indexing, *Signal Processing*, Vol. 90, No. 1, pp. 331-343, January, 2010.

[19] N. M. Makbol, B. E. Khoo, Robust Blind Image Watermarking Scheme Based on Redundant Discrete Wavelet Transform and Singular Value Decomposition, *AEU-International Journal of Electronics and Communications*, Vol. 67, No. 2, pp. 102-112, February, 2013.

[20] Y. K. Lin, A Data Hiding Scheme Based upon DCT Coefficient Modification, *Computer Standards & Interfaces*, Vol. 36, No. 5, pp. 855-866, September, 2014.

[21] X. Zhang, S. Wang, Z. Qian, G. Feng, Self-Embedding Watermark with Flexible Restoration Quality, *Journal Multimedia Tools and Applications*, Vol. 54, No. 2, pp. 385-395, August, 2011.

[22] C. F. Lee, H. L. Chen, A Novel Data Hiding Scheme Based on Modulus Function, *Journal of Systems and Software*, Vol. 83, No. 5, pp. 832-843, May, 2010.

[23] M. H. Mohamed, N. M. Al-Aidroos, M. A. Bamatraf, Data Hiding Technique Based on LSB Matching towards High Imperceptibility, *MIS Review*, Vol. 18, No 1, pp. 57-69, September, 2012.

[24] Z. Xia, X. Wang, X. Sun, Q. S. Liu, N. Xiong, Steganalysis of LSB Matching, *Multimedia Tools and Applications*, Vol. 75, No. 4, pp. 1947-1962, February, 2016.

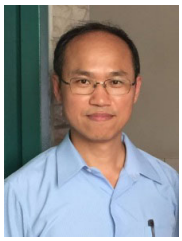
[25] T. Kim, Side Match and Overlap Match Vector Quantizers for Images, *IEEE Transactions on Image Processing*, Vol. 1, No. 2, pp. 170-185, April, 1992.

[26] Y. D. Qu, C. S. Cui, S. B. Chen, J. Q. Li, A Fast Subpixel Edge Detection Method Using Sobel-Zernike Moments Operator, *Image and Vision Computing*, Vol. 23, No. 1, pp. 11-17, January, 2005.

[27] X. Zhang, Z. Qian, Y. Ren, G. Feng, Watermarking with Flexible Self-Recovery Quality Based on Compressive Sensing and Compositive Reconstruction, *IEEE Transactions on Information Forensics and Security*, Vol. 6, No. 4, pp. 1223-1232, December, 2011.

[28] X. Zhang, S. Wang, Z. Qian, G. Feng, Reference Sharing Mechanism for Watermark Self-embedding, *IEEE Transactions on Image Processing*, Vol. 20, No. 2, pp. 485-495, February, 2011.

## Biographies



**Yung-Chen Chou** received the MS degree in Information Management from Chaoyang University of Technology, Taichung, Taiwan, 2002. He received Ph.D degree in Computer Science and Information Engineering 2008 from National Chung-Cheng University, Chiayi, Taiwan. He has been an associated professor of Asia University, Taichung, Taiwan. His research interests: steganography, image processing.

**Chun-Hsiu Yeh** received the Ph.D. degree from National Chung Hsing University in 2017, and the master degree from Chaoyang University of Technology in 2001. Her research interests include image processing, image restore. Now, she is a assistant professor with the Department of Management Information, Chung Chou University of Science and Technology.

**Jau-Ji Shen** received his Ph.D. degree from National Taiwan University in 1988, and the master degree from National Chung Hsing University in 1984. His research interests include digital image, software engineering, information security. Now, he is a professor with the Department of Management Information Systems, National Chung Hsing University.

**Jinn-Ke Jan** received the B.S. degree in physics from Catholic Fu Jen University in 1974 and the M.S. degree in information and computer science from the University of Tokyo in 1980. He is a professor with the department of Computer Science & Engineering, National Chung Hsing University. His research interests include computer cryptography, data structures.