Sharing a verifiable secret image using two shadows

Chin-Chen Chang\textsuperscript{a,b}, Chia-Chen Lin\textsuperscript{c,∗}, T. Hoang Ngan Le\textsuperscript{d}, Hoai Bac Le\textsuperscript{d}

\textsuperscript{a}Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan, ROC
\textsuperscript{b}Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan, ROC
\textsuperscript{c}Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan, ROC
\textsuperscript{d}Department of Computer Science, Natural Science University, 227 Nguyen Van Cu, District 5, HCMC, Vietnam

\textbf{ARTICLE INFO}

Article history:
Received 15 September 2008
Received in revised form 16 February 2009
Accepted 20 April 2009

Keywords:
Visual secret sharing
Error diffusion
Inverse halftoning
\cite{7, 4} Hamming code
Verifiable secret image
Clustering

\textbf{ABSTRACT}

In this paper, we propose a novel (2, 2) verifiable secret sharing (VSS) scheme, which not only protects a secret image but also allows users to verify the restored secret image in the revealing and verifying phase, for all binary, grayscale and color images. Error diffusion and image clustering techniques are adopted to achieve our objective. Experimental results and discussions show that the proposed scheme, with its smaller shadow size and lower computational complexity, obviously outperforms previous VSS schemes designed either with or without the cheating prevention mechanism. Moreover, the use of a halftone logo gives an efficient solution to verifying whether the restored secret image is correct by using a halftone logo.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Unlike data hiding technology, which hides secret information in innocuous carriers, secret sharing takes a different approach to ensure information protection and secure transmission. In contrast to data hiding technology with which the secret information is hidden and reconstructed by only one receiver, secret sharing technology allows the secret data $D$ to be divided into $n$ pieces. In other words, the secret data $D$ can be held by $n$ participants.

A secret sharing (SS) scheme, called a $(k,n)$ threshold scheme, allows $n$ participants to share secret data. Each participant holds a part of the secret data, called a shadow or a share, and each shadow reveals no information about the original secret image. To reconstruct the secret data, there must be complete knowledge of at least $k$ shadows, where $k \leq n$. This means that any $k$ out of $n$ participants must share their own shadows among themselves, then individually reconstruct the secret data. Two well-known $(k,n)$ SS schemes were first independently proposed in 1979 by George Blakley \cite{2} and Adi Shamir \cite{24}. In Shamir’s scheme \cite{24}, a prime number $g$ is randomly chosen and the polynomial sharing function of degree $k-1$ is defined as $f(x) = (a_0 + a_1x + \cdots + a_{k-1}x^{k-1}) \mod g$ to divide $D$ into $n$ shadows, where $a_1, a_2, \ldots, a_{k-1}$ are random numbers and $a_0 = D$. Each $D_i$ for $i = 1 \text{ to } n$ can be derived by $D_i = f(i)$, where $i = 1 \text{ to } n$, and each one of $D_1, D_2, \ldots, D_n$ is treated as a shadow. To obtain the secret data $D$, any $k$ or more shadows of $D_i$’s can reconstruct $f(x)$. All coefficients of the polynomial function $f(x)$ can be derived from Lagrange’s interpolation formula, and finally the secret data $D = a_0 = f(0)$ can also be calculated.

In 1995, Naor and Shamir \cite{21} extended the secret sharing scheme from the number domain to the image domain, and this kind of secret sharing scheme is called a visual secret sharing (VSS) scheme. Over the past decade, many improved VSS schemes have been proposed to reduce the computational cost while maintaining acceptable quality of the reconstructed secret image \cite{1, 3, 10, 15, 16, 21}. Some of them \cite{1, 3, 15, 21} adopt a stacking approach to generate the reconstructed image during the revealing phase. Their methods for reconstructing a secret image are simple but bring with them the pixel expansion problem. Others solve the pixel expansion problem but incur the cost of high computational complexity \cite{10, 16}. To overcome pixel expansion and high computational complexity at the same time, Yang \cite{31}, Cimato et al. \cite{12}, and Wang et al. \cite{30} introduced the probabilistic concept into a new class of VSS schemes called ProbVSS schemes. These schemes involve two Boolean operations, XOR (\texttt{+}) and AND (\texttt{&}), for binary images. In 2008, Chang et al. \cite{7, 8} extended the application of ProbVSS to encompass both grayscale images and color images, respectively.

Even though a variety of secret sharing schemes have been proposed, scholars initially evaluate the performance of each secret sharing scheme based on four general criteria: security, accuracy, computational complexity and shadow size. Not until Horng et al.
proposed two schemes to prevent cheating [15] in 2006 did the cheating prevention issue gain scholars’ attention. One of Horng et al.’s two schemes involves authentication based on additional shares or images, which are required to verify the integrity of each participant’s share. The other is designed to make it harder for cheaters to predict the structure of transparencies held by honest, legal participants. Later, Prisco and Santis proposed a (2, n) threshold scheme for binary images that is robust against any cheaters without requiring the use of extra shares [23]. In Prisco and Santis’ scheme, a cheater group cannot force an honest participant to reconstruct an incorrect secret image during the revealing phase. Because the secret images discussed in Horng et al. [15] and Prisco and Santis [23] are binary images, in 2007 Zhao et al. [33] proposed a verifiable secret sharing scheme for grayscale images based on an improved version of Thien and Lin’s scheme [27]. Zhao et al.’s scheme allows honest participants to identify cheaters using extra information. However, in Zhao et al.’s scheme the dealer must publish two parameters and each participant must publish its secret share reference. That is to say, the dealer and participants must communicate among themselves to verify each unique secret share reference, which is derived by a secret share. Moreover, the amount of extra information is increased if their scheme extends the application domain from grayscale images to color images, and they cannot be directly applied to binary images. In contrast with these schemes, Chang et al. [9] and Yang et al. [32] embed the generated shadows and authentication codes into the cover image so that the extracted authentication codes can be used to check integrity of the stego-image [26]. To sum up, most existing verifiable secret sharing schemes allow participants to verify received shadows rather than a reconstructed secret image. When using such cheating prevention approaches, participants must verify their received shadow as soon as they arrive from the dealer during the shares construction phase, then repeat the verification procedure when they collect shadows from other participants during the revealing phase. In other words, the verification mechanism must be performed twice. Because the ultimate objective of shadow verification is to make sure the reconstructed secret image is correct, is there any approach that achieves the same objective without requiring participants to perform shadow verification during both the shares reconstruction phase and the revealing phase? To answer this question, in this paper we propose a (2, 2) VSS scheme that allows honest participants to verify the reconstructed secret image, thus enhancing their confidence in the reconstructed secret image. Moreover, to increase the practicability of the proposed scheme, a Boolean operation is used to produce shadow set and generate the reconstructed secret image. In this way, the computational complexity of the proposed verifiable secret image scheme remains low. To achieve our objectives, three techniques are utilized in this scheme. The first is error diffusion technique [25] helping to transform a grayscale image into a binary image; the second is edge lookup inverse halftoning (ELIH) technique [11] relied on edge detection and a lookup table to generate a reconstructed grayscale image from a halftone image; clustering image technique using a (7, 4) Hamming coding is the last. The experimental results confirm that our scheme not only satisfies four basic criteria: security, accuracy, computational complexity and the size of shadow, but also overcomes the disadvantages being caused by any dishonest participants. Moreover, the proposed scheme can be applied not only to natural images including grayscale and color images, but also to binary images with lightly modification. The related modification description for working on binary images is discussed further in Section 3.

The rest of this paper is organized as follows: Section 2 briefly describes three techniques: error diffusion technique, ELIH technique, and clustering based on (7,4) Hamming coding. A detailed description of our proposed scheme is in Section 3. Section 4 demonstrates the experimental results of our scheme, with an analysis of performance on binary, grayscale and color images. Finally, we make conclusions in Section 5.

2. Related works

2.1. Error diffusion technique

Error diffusion is the most popular technique, which uses bi-level pixels to simulate a gray-scale image or a color image [5,18,25]. Given a pixel \( G(x,y) \) at the position \( (x,y) \) in the grayscale image \( G \), \( G(x,y) \) becomes \( H(x,y) \) whose value is either 0 or 255 after a quantization procedure. During the quantization procedure, a threshold \( T \) is used to determine \( H(x,y) \) of an input pixel \( G(x,y) \) to be 255 or 0. The quantization error \( e(x,y) \) is computed as \( e(x,y) = G(x,y) - H(x,y) \) of the current pixel is diffused by the error filter. The signal consisting of past error values is passed through the error filter to produce a correction factor that is added to future input pixels. If the quantization error is negative, \( G(x,y) \) is quantized to be 255 so that the corresponding \( H(x,y) \) is set as 255. In this case, its neighboring pixels values must be decreased in such a manner that locally the errors cancel out. In contrast, the value of \( G(x,y) \) is quantized to 0 and its neighboring pixels values will be increased. Fig. 1 shows a flowchart of the error diffusion technique.

The first error diffusion method proposed by Floyd and Steinberg [13], is adopted here to transform a grayscale image into a binary image. Assume an original grayscale \( G \) whose size is \( w \times h \) and each pixel is denoted as \( G(x,y) \), where \( 1 \leq x \leq h \) and \( 1 \leq y \leq w \). The threshold \( T \) is set as 127. The following steps are used to create a binary image \( H \) from a grayscale image \( G \):

**Step 1:** Set \((x,y) = (1,1)\), that is, consider the first pixel.

**Step 2:** Compute the quantization error value \( e(x,y) \) and set the value of pixel \( H(x,y) \) as 0 or 255 according to

\[
e(x,y) = G(x,y) - H(x,y),
\]

\[
H(x,y) = \begin{cases} 
255 & \text{if } G(x,y) \geq T, \\
0 & \text{otherwise.}
\end{cases}
\]

**Step 3:** Diffuse the quantization error \( e(x,y) \) over the neighboring pixels of \( G(x,y) \) and update the grayscale image \( G \). The four neighboring pixels: \( G(x+1, y), G(x, y+1), G(x+1, y+1) \) are altered. These values are modified by using Eq. (3). Fig. 2 illustrates

![Flowchart of error diffusion architecture.](image)

![The kernel weight of Floyd and Steinberg's error filter.](image)
the kernel weight of Floyd and Steinberg’s error filter.
\[ G(x, y + 1) = G(x, y + 1) + 7 \times e(x, y)/16, \]
\[ G(x + 1, y) = G(x + 1, y) + 5 \times e(x, y)/16, \]
\[ G(x + 1, y - 1) = G(x + 1, y - 1) + 3 \times e(x, y)/16, \]
\[ G(x + 1, y + 1) = G(x + 1, y + 1) + 1 \times e(x, y)/16. \]  
(3)

Step 4: Check for whether the processed pixel’s \( x = h \) and \( y = w \); if it does not satisfy, input the next pixel and then perform Step 2. Otherwise, the procedure is terminated and a binary image \( H \) is generated.

### 2.2. Edge lookup inverse halftoning (ELIH) technique

The inverse halftoning technique is used to generate a reconstructed grayscale image from an input halftone image. ELIH, proposed by Chung and Wu [11] in 2005, uses a lookup table (LUT) technique for inverse halftoning. Experimental results proved Chung and Wu’s ELIH successfully enhances the quality of the reconstructed grayscale image. Fig. 3 is a flowchart briefly representing the ELIH technique.

Assume the training data set is a set of \( n \) training image pairs \((G_i, H_i)\), where \( G_i \) and \( H_i \) are denoted as the \( i \)th grayscale image \( G_i \) and the corresponding halftone image \( H_i \).

**Step 1:** Generate the lookup table (LUT). The procedure for LUT generation is illustrated in Fig. 4.

**Step 1.1:** Set \( i \) as 1, meaning that the first pair of training images \((G_i, H_i)\) is considered. Set lookup table \(|LUT| = 0 \). Here, the array \(|LUT| \) is used to map the input halftone image to the base grayscale image. The \(|LUT| \) array will be used in Step 2.

**Step 1.2:** Divide images \( H_i \) and \( G_i \) into overlapping 4×4 blocks, respectively. Note that \( H_{ij} \) and \( G_{ij} \) are the \( j \)th halftone block of \( H_i \) and the \( j \)th grayscale block of \( G_i \), respectively.

**Step 1.3:** Calculate index \( I_{ij} \) for each halftone block \( H_{ij} \), and update the value of the intermediate \( LUT[I_{ij}] \) by using Eqs. (4) and (5), respectively. Note that each grayscale blocks \( G_{ij} \) is represented by one pixel at the position \((3, 3)\) called \( G_{ij}(3, 3) \).

\[ I_{ij} = \sum_{k=1}^{4} \sum_{l=1}^{4} 2^{4-l-1} \times H_{ij}(k, l), \]  
(4)

\[ LUT[I_{ij}] = LUT[I_{ij}] + G_{ij}(3, 3). \]  
(5)

**Step 2:** After applying Step 1 to a set of \( n \) training images in succession, retrieve a set of reconstructed grayscale images called \( I_{Gi} \), for \( i = 1, 2, \ldots, n \). Apply the Canny edge detector to each reconstructed grayscale image \( I_{Gi} \) to generate an edge map \( M_i \), for \( i = 1, 2, \ldots, n \). Each edge map \( M_i \) consists of a set of 4×4 blocks. Therefore, the \( j \)th block of the edge map \( M_i \) is denoted as \( M_{ij} \). Combine the lookup table \( LUT \) generation procedure described in Step 1 with a set of 4×4 edges, called an ELUT, is generated. The order of the edge pattern of the \( j \)th block is denoted as \( M_{ij} \). The index value for \( j \)th block is \( I_{ij} \) for \( 0 \leq I_{ij} < 2^{4} - 1 \). Finally, the mean grayscale value is derived from \( ELUT[I_{ij}, M_{ij}] \), where \( 0 \leq M_{ij} \leq 38 \) based on the number of edge patterns reported by Chung and Wu [11]. Note that in Chung and Wu’s scheme, the value of \( G_{ij}(3, 3) \) is computed as \( G_{ij}(3, 3) = ELUT[I_{ij}, M_{ij}] \).

![Flowchart of ELIH technique.](image-url)

![Procedure for the LUT generation.](image-url)
2.3. Image clustering by \((7, 4)\) Hamming coding

In this paper, we design a clustering method based on \((7, 4)\) Hamming coding to classify the non-overlapping blocks of a halftone image \(HI\) into groups. In this section, we briefly describe how to use \((7, 4)\) Hamming coding to perform clustering on the non-overlapping blocks of a halftone image \(HI\) [4]. Let \(HI\) and \(P(7, 4)\) be a halftone image and a \((7, 4)\) Hamming coding, respectively [20]. Cluster processing over a halftone image \(HI\) can be done according to the following steps.

Step 1: Initiate \(GR[\cdot] = 0\) and \(NR[\cdot] = 0\), where the array \(GR[\cdot]\) is a set of groups and \(NR[\cdot]\) is the number of elements in each group. Each group in the \(GR[\cdot]\) is empty at the beginning. The \(P(7, 4)\) used in this scheme is set as

\[
P(7, 4) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.
\]

Step 2: Divide \(HI\) into the non-overlapping blocks, each block contains 7 bits. Note that \(HI_z\) represents the \(z\)th block of \(HI\).

Step 3: Set \(z = 1\), that is, consider the first block to be processed.

Step 4: After \(HI_z\) is multiplied by \(P(7, 4)\), the values of \(GR[\cdot]\) and \(NR[\cdot]\) are updated by using Eq. \((6)\). Let \(t\) be the index and be calculated by transforming vector \(P(7, 4) \times (HI_z)^T\) into a decimal number and ranges from 0 to 7. Note that \((HI_z)^T\) is a transpose vector of \(z\)th block \(HI_z\) in the halftone image \(HI\).

\[
NR[t] = NR[t] + 1,
\]

\[
GR[t] = GR[t] \cup HI_z.
\]

Step 5: If the current block is the last block in \(HI\), the procedure is terminated and the results of \(NR[\cdot]\) and \(GR[\cdot]\) are outputted as the clustering results. Otherwise, go to the next block in halftone image \(HI\) by increasing \(z\) by one, then perform Step 4.

**Fig. 5. Procedure of reconstructed grayscale image generation in Step 4.**

**Table 1**

<table>
<thead>
<tr>
<th>(H_t)</th>
<th>(T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>([0 0 0 1 1 1)</td>
<td>0</td>
</tr>
<tr>
<td>([0 0 0 1 0 1)</td>
<td>2</td>
</tr>
<tr>
<td>([0 1 0 0 1 0)</td>
<td>4</td>
</tr>
<tr>
<td>([0 1 0 1 0 0)</td>
<td>6</td>
</tr>
</tbody>
</table>

**Table 2**

Clustering results for a set of four 7-bit blocks based on their \(t\) values.

**The following example demonstrates the five steps above. Let us assume that the halftone image is**

\[
HI = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.
\]

**Step 1:** \(GR[\cdot] = \), \(NR[\cdot] = 0\) and

\[
P(7, 4) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.
\]

**Step 2:** Because \(HI\) is a halftone image sized 7×4 pixels, after division into a set of non-overlapping 7-bit blocks, we get four blocks: \(H_{11} = [0 0 0 1 1 1 1]\), \(H_{12} = [0 0 0 1 0 1 0]\), \(H_{31} = [0 1 0 0 1 0 1]\) and \(H_{41} = [1 0 1 1 0 0 0]\).

**Step 3:** Set \(z = 1\). Take the first block \(H_{11} = [0 0 0 1 1 1 1]\) into consideration.

**Step 4:** Compute \(t = P x (H_{11})\) as

\[
\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.
\]
Update the values for $GR[]$ and $NR[]$ by using Eq. (6) as follows: $NR[0] = 1$; $GR[0] = \{0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1\}$.

Steps 2–4 are performed repeatedly until all block in the halftone image $HI$ have been processed. Here, we have four blocks in our halftone image, the $t$ value for each 7-bit block and the clustering results are demonstrated in Tables 1 and 2, respectively [6,14]. From Table 2, we can see that in this example four blocks of halftone images have been successfully classified into three groups. For example, $HI_1$ and $HI_3$ belong to group $GR[0]$.

### 3. Proposed verifiable secret sharing scheme for grayscale images

In the proposed scheme, we use a halftone logo to ascertain the exactitude of the reconstructed secret image so that honest participants do not need to perform twice verification procedures as most VSS scheme with cheating prevention mechanism. Our scheme consists of two phases: (1) shares construction phase and (2) revealing and verifying phase. Because our proposed scheme is (2,2) VSS, dealer generates two shadows from an original secret grayscale image called $GI$ and a halftone logo called $HL$, respectively, during the shares construction phase. Revealing and verifying phase assists participants to reconstruct a grayscale secret image $GI$ and extract the halftone logo $HL$ from the collected shadows. If participants need sounder proof, they also can ask the dealer to announce the original halftone logo. Note that in this paper we assume the dealer is honest in our paper. The detailed descriptions are given in the following subsections.

#### 3.1. Shares construction phase

The shares construction phase consists of six steps as illustrated in the flowchart in Fig. 6. In the shares construction phase, Floyd and Steinberg’s error diffusion technique [13] is used to transform a grayscale image into a halftone image. Assume that $GI$ contains 7×X grayscale pixels in size and $HL$ contains 3×X bi-level pixels in size. In practical applications, this condition can be satisfied by padding operations.

**Step 1:** Apply Floyd and Steinberg’s error diffusion technique to the secret grayscale image $GI$. This technique transforms the secret grayscale image $GI$ into the secret halftone image $HI$. Meanwhile, the halftone image $HL$ is maintained. After this step is completed, one halftone logo $HL$ and one secret halftone image $HI$ are obtained.

**Step 2:** Halftone image $HI$ is clustered into eight groups by applying the (7,4) Hamming code-based clustering method described in Section 2.3. Let $GR[\cdot]$ and $NR[\cdot]$, for $t = 0, 1, \ldots, 7$, be the array of eight groups and the number of elements in each group, respectively.

**Step 3:** Divide halftone image $HI$ into non-overlapping blocks, each block containing 7 pixels and each pixel having only one bit with a value of either ‘0’ or ‘1’. Our scheme is based on two assumptions: (1) the grayscale image $GI$ has 7×X pixels and each pixel contains 8 bits, and (2) the halftone image $HI$ has 7×X pixels and each pixel only contains only 1 bit. Note that if the pixel numbers of $GI$ and $HI$ cannot be divided by 7, “0” is added to make them satisfy. Therefore, a set of X non-overlapping blocks is divided from halftone image $HI$. Note that $BL_t$ represents the $t$th block of the halftone image $HI$.

Let us give an example where the value of $X$ is 4 and the grayscale image $GI$ is set as (Fig. 7)

$$
\begin{bmatrix}
0 & 0 & 0 & 255 & 255 & 255 & 255 \\
0 & 0 & 0 & 255 & 255 & 255 & 0 \\
0 & 255 & 0 & 0 & 255 & 0 & 255 \\
255 & 0 & 255 & 0 & 255 & 0 & 0
\end{bmatrix}
$$

**Step 4:** Divide halftone logo $HL$ into non-overlapping blocks and each block contains three pixels. Based on the same assumption mentioned in Step 3, halftone logo $HL$ consists of 3×X bi-level pixels; therefore, a set of X non-overlapping blocks is divided from halftone logo $HL$. Note that $BL_1$ is the $t$th block of $HL$. Let us assume the value of $X$ is 5 and the grayscale image $HI$ is set as

$$
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 0 \\
1 & 1 & 0 \\
0 & 0 & 0
\end{bmatrix}
$$

After dividing the halftone logo $HL$ into non-overlapping blocks, we obtain a set of four blocks: $BL_1 = \{0 \ 1 \ 0\}$, $BL_2 = \{0 \ 0 \ 0\}$, $BL_3 = \{1 \ 1 \ 0\}$, and $BL_4 = \{0 \ 0 \ 0\}$.

**Step 5:** Construct the first shadow by using eight clustered groups $GR[]$, a set of X blocks $BL_1$ and $X$ blocks $BL_2$, where $z = 0, 2, \ldots, X$. Step 5 can be broken down into three substeps.

**Step 5.1:** Set $z = 1$, which means that blocks $BL_1$ and $BL_1$ are considered.

**Step 5.2:** Convert the binary value of block $BL_2$ into a decimal value. Let $t$ be the decimal value of block $BL_2$, $t = 0, 1, \ldots, 7$. The most different element in $GR[]$ of $BL_3$ called $BS_t$ is an element contained in $GR[\cdot]$ so that the Hamming distance between elements $BL_2$ and $BS_t$ is the largest.

**Step 5.3:** If $z$ is < $X$, $z$ is increased by one; go to Step 5.2. Otherwise if $z$ is equal to $X$, all blocks in the halftone image $HI$ and the halftone logo $HL$ are considered. Thus, the first shadow is generated by concatenating a set of $X$ blocks $BS_1$.

**Step 6:** To construct a random noise-like image for each shadow to make sure no shadow leaks any information about the secret image, all pixels in halftone image $HI$ are permuted by using

---

**Fig. 6.** Flowchart of the shares construction phase.
Grayscale image $GI$

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>255</th>
<th>255</th>
<th>255</th>
<th>255</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>255</td>
</tr>
<tr>
<td>255</td>
<td>0</td>
<td>255</td>
<td>255</td>
<td>255</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Halftone image $HI$

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Divide

$BL_1$ = 0 0 0 1 1 1 1
$BL_2$ = 0 0 0 1 0 1 0
$BL_3$ = 0 1 0 0 1 0 1
$BL_4$ = 1 0 1 1 0 0 0

Fig. 7. Example of Step 3 of the shares construction phase.

2D-Torus Automorphism transformation [29]. A new position $(x', y')$ of pixel $HI(x, y)$ is calculated by using

$$(x', y') = (x, y) \times \begin{bmatrix} 1 & 1 \\ k & k+1 \end{bmatrix} \mod A,$$  

(7)

where $k$ is a private key and $A$ is the number of blocks. The details of 2D-Torus Automorphism transformation are mentioned in Voyatzis et al.'s algorithm [29].

Finally, the second shadow is generated by combining the first shadow and the permuted halftone image using the XOR operation. The following example gives a better explanation of the shares construction phase. Following values of the parameters $X, GI$, and $HL$ are chosen as 4.

$$GI = \begin{bmatrix} 0 & 0 & 0 & 255 & 255 & 255 & 255 \\ 0 & 255 & 0 & 0 & 255 & 0 & 255 \\ 255 & 0 & 255 & 255 & 0 & 0 & 0 \end{bmatrix},$$

and

$$HL = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

given above. Steps in the shares construction phase are demonstrated as follows:

Step 1: Apply Floyd and Steinberg’s error diffusion technique to $GI$ so that a halftone image

$$HI = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

is obtained.

Step 2: Cluster the halftone image $HI$ by applying the $(7, 4)$ Hamming, we obtain $GR[1]$ and $NR[r]$ with values of: $NR[0] = 2$; $NR[1] = 0$; $NR[2] = 1$; $NR[3] = 0$; $NR[4] = 0$; $NR[5] = 0$; $NR[6] = 1$; $NR[7] = 0$; $GR[0] = \{0, 1, 0, 0, 1, 0, 1\}; \{0, 0, 0, 0, 1, 1, 1\}; GR[1] = \{1\}; GR[2] = \{0, 0, 0, 1, 0, 1\}; GR[3] = \{\}; GR[4] = \{\}; GR[5] = \{\}; GR[6] = \{0, 1, 0, 0, 0, 0\}; GR[7] = \{\}$.

The clustering results are obtained from Section 2.3.

Step 3: The set of $X(X = 4)$ non-overlapping blocks divided from the halftone image $HI$ is: $BL_1 = [0 0 0 1 1 1 1]$, $BL_2 = [0 0 0 1 0 1 0]$, $BL_3 = [0 1 0 0 1 0 1]$ and $BL_4 = [1 0 1 1 0 0 0]$.

Step 4: The set of $X(X = 4)$ non-overlapping blocks divided from the halftone logo $HL$ is: $BL_1 = [0 1 0 0, BL_2 = [0, 0, 0], BL_3 = [1 1 0]$ and $BL_4 = [0, 0, 0]$.

Step 5.1: Set $z = 1$. Take the first block $BL_1 = [0 1 0 0 1 0 1]$ and $BL_1 = [0 0 0]$ into consideration.

Step 5.2: The decimal value of $BL_1$ is 2. Look at the $GR[2]$ to find $BS_1$. There is one element in $GR[2]$. Therefore, $BS_1 = [0 0 0 1 0 1 0]$. Step 5.2 is repeatedly performed until $BS_1, BS_2$ and $BS_4$ are processed and the corresponding $BS_2 = [0 0 0 1 1 1 1], BS_3 = [1 0 1 1 0 0 0]$ and $BS_4 = [0 1 0 0 1 0 1]$ are generated.

Step 5.3: The first shadow is created by concatenating $BS_1$, $BS_2$, $BS_3$ and $BS_4$ as

$S_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$.

Step 6: Assume that we ignore the permutation operation here. The second shadow is created by the XOR operation between the first shadow and the halftone image $HI$. Apply the XOR operation to the first shadow

$S_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$.

3.2. Revealing and verifying phase

Our proposed revealing and verifying phase can achieve two objects. One is to reconstruct the halftone logo $HL$ and the halftone
image \(H'_L\) and then convert the halftone image \(H'_L\) into the reconstructed grayscale image \(G'_I\). The other is to assist the honest participants to verify the reconstructed grayscale image \(G'_I\) based on the reconstructed halftone logo \(H'_L\). The flowchart in Fig. 8 illustrates the procedure for the revealing and verifying phase.

Step 1: The first shadow \(S_1\) is divided into the non-overlapping blocks having seven bi-level pixels in size.

Step 2: Each block of 7 pixels of \(S_1\) is multiplied by \(P(7,4)\) which is a \((7,4)\) Hamming coding. Thus, we can obtain three bits from each block. Based on \(X\) blocks divided from the \(7 \times 7\) pixels in \(S_2\), we attain a set of \(X\) blocks, and each block consists of three bits. By combining these \(X\) blocks together, we can reconstruct the halftone logo \(HL'_1\).

Step 3: Authenticate the reconstructed secret image based on the condition of the reconstructed halftone logo \(HL'_1\). Because we assume the dealer is honest, if the reconstructed grayscale image \(GI'_1\) is a readable rather than a noise-like image, the reconstructed secret image is authenticated. If participants need sounder proof, they can ask the dealer to announce the real halftone logo, so that the reconstructed secret image is authenticated. If participants need sounder proof, they can ask the dealer to announce the real halftone logo, so that the reconstructed secret image is authenticated. If participants need sounder proof, they can ask the dealer to announce the real halftone logo, so that the reconstructed secret image is authenticated. If participants need sounder proof, they can ask the dealer to announce the real halftone logo, so that the reconstructed secret image is authenticated. If participants need sounder proof, they can ask the dealer to announce the real halftone logo, so that the reconstructed secret image is authenticated. If participants need sounder proof, they can ask the dealer to announce the real halftone logo, so that the reconstructed secret image is authenticated. If participants need sounder proof, they can ask the dealer to announce the real halftone logo, so that the reconstructed secret image is authenticated. If participants need sounder proof, they can ask the dealer to announce the real halftone logo, so that the reconstructed secret image is authenticated. If participants need sounder proof, they can ask the dealer to announce the real halftone logo, so that the reconstructed secret image is authenticated.

Step 4: We need to de-permute the second shadow \(S_2\) using 2D-Torus Automorphism transformation. The halftone image \(H'_L\) is then created by performing the XOR operation on shadows \(S_1\) and \(S_2\). Here, \(H'_L\) is a binary image consisting of \(7 \times 7\) bi-level pixels.

Step 5: The inverse halftoning transform mentioned in Section 2.2 is applied into the halftone image \(H'_L\) to generate the reconstructed original grayscale image \(G'_I\).

To illustrate this phase more clearly, the following example is a continuation of the example given to illustrate the shares construction phase. To obtain the reconstructed secret image, the above steps are performed on the first shadow

\[
S_1 = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1
\end{bmatrix}
\]

and the second shadow

\[
S_2 = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1
\end{bmatrix}
\]

Flowchart for the revealing and verifying phase.

Fig. 8. Flowchart for the revealing and verifying phase.
Input: Binay image $HI$ and halftone logo $HL$

**Step 1:** Image clustering

**Step 2:** Divide into non-overlapping blocks

**Step 3:** Divide into non-overlapping blocks

**Step 4:** Choose the most different element in $GR[]$

**Step 5:** Permute

Reconstructed halftone logo $HL'$

Reconstructed halftone image $HI'$

Shadow $S_2$

Shadow $S_1$

---

**Fig. 9.** Flowchart of the shares construction phase for a binary image.

 Shadow $S_1$ → **Step 1:** Divide into non-overlapping blocks → **Step 2:** Image clustering → Reconstructed halftone logo $HL'$ → **Step 3:** Verify the reliability → **Step 4:** Perform XOR operation → YES → **Step 3:** Verify? → NO → Do nothing

Fig. 10. Flowchart of revealing and verifying phase for a binary image.

and the second shadow

$$S_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

to obtain the reconstructed halftone image

$$HI' = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

**Step 5:** Apply the ELIH technique mentioned in Section 2.2 to the reconstructed halftone image $HI'$, the reconstructed grayscale image

$$GI' = \begin{bmatrix} 0 & 5 & 0 & 200 & 255 & 255 & 255 \\ 41 & 0 & 0 & 255 & 0 & 200 & 10 \\ 0 & 255 & 0 & 0 & 245 & 25 & 197 \\ 250 & 15 & 224 & 255 & 10 & 0 & 0 \end{bmatrix}$$

is obtained.

We can extend this scheme to binary and color images. With a binary image, we make a slight modification by omitting the first step in the shares construction phase and omitting the last step in the revealing and verifying phase. Figs. 9 and 10 show the flowcharts for the two phases for the binary images, respectively.

In the shares construction phase, the halftone image $HI$ is considered the input binary image. All the steps remain except the first step for grayscale images during the shares construction phase. Thus, the $s$th step in the flowchart corresponding to the grayscale image becomes the $(s-1)$th step in this flowchart. As for the revealing and verifying phase of a binary image, it is generated by omitting the last step from that of the grayscale image processing shown in Fig. 10.

We consider a color image to be a collection of three grayscale images corresponding to the Red, the Green, and the Blue planes. Each color plane is treated as one grayscale image. A flowchart showing how to generate two shadows from one color image is shown in Fig. 11. Each shadow is generated by composing the shadows from each plane. Namely, the first shadow $S_1$ is generated by composing the first shadows $SR_1, SG_1, SB_1$ from the Red, Green, and Blue planes. The second shadow $S_2$ is generated by composing the second shadows $SR_2, SG_2, SB_2$ from the Red, Green, and Blue planes.

**4. Experimental results and analysis**

The experimental results demonstrate two objectives. The first is to prove that our proposed scheme satisfies four general criteria: security, accuracy, computational complexity and shadow size, also called pixel expansion. The second is to demonstrate that confidence in a set of shadow as well as the reconstructed secret image can be verified by comparing the original halftone logo with the extracted halftone logo. Experimental results and discussions related to the four general criteria are presented in Sections 4.1–4.4. A demonstration of our cheating prevention function is given in Section 4.5.
prove that our scheme outperforms other schemes that use a cheating prevention mechanism, some discussions and comparisons are given in Section 4.6.

Experiments were conducted on a PC with an Intel(R) Core™ 2 CPU 1.83 GHz and a 1-GB RAM. The operating system is Windows XP Professional, and our algorithms were programmed by Matlab 7.0. To illustrate that our scheme can be performed well on both binary, grayscale and color images, we used three sets of test images sets shown in Fig. 12(1)–(3), respectively. The first set contains four 128×128 binary images, “Lena”, “Jet”, “Pepper”, and “Sailboat”. The second set contains four 128×128 grayscale images, “Sailboat”, “Tiffany”, “Peppers”, and “Lena”. The other set contains four 128×128 color images, “Barbara”, “Jet”, “Watch”, and “Zelda”. The two halftone logos used to verify secret data, “CCU” and “MSN”, are shown in Tables 3–5, which correspond to the binary, grayscale and color test images.

From Tables 3–5, we can see that each shadow is a noise-like halftone image; therefore, the security of the proposed scheme is guaranteed when subjected to the human visual system.

4.1. Security

We set out to prove that our scheme satisfies the security criterion; that is, preventing a shadow from leaking any information about the original secret image. The sets of shadows generated by our scheme are shown in Tables 3–5, which correspond to the binary, grayscale and color test images.

To further test the influence of a one-pixel change on each shadow and prove that our scheme can withstand a differential attack, we used two common measures: Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI). The first measure, NPCR, determines the percentage of different pixel numbers between the two shadows, $S_1$ and $S_2$, whereas UACI measures the average intensity of differences between the two shadows. NPCR is defined in Eq. (8).

$$\text{NPCR}(S_1, S_2) = \frac{\sum_{i=1}^{w} \sum_{j=1}^{h} D_{ij}}{w \times h} \times 100\%,$$

where $w$ and $h$ are the width and height of shadows $S_1$ and $S_2$ and $D_{ij}$ is defined as

$$D_{ij} = \begin{cases} 1, & S_{1ij} \neq S_{2ij}, \\ 0, & S_{1ij} = S_{2ij}. \end{cases}$$

Here, $S_{1ij}$ and $S_{2ij}$ are the $i$th row and $j$th column pixel of same sized images $S_1$ and $S_2$, respectively.

According to Kwoka, and Tang’s report [17], for two random images, the expected NPCR value is $(1 - 2^{-L}) \times 100\%$, where $L$ is the number of bits used to present one pixel of an image (at here the image is a shadow). In our scheme, when secret images are binary and grayscale images, each pixel of the shadow has one bit. Therefore, the expected NPCR value is $\text{NPCR} = (1 - 2^{-1}) \times 100\% = 50\%$. When secret images are color images, each pixel of the shadow has three bits. The expected NPCR value is $\text{NPCR} = (1 - 2^{-3}) \times 100\% = 87.5\%$. As for UACI, it is determined by

$$\text{UACI}(A, B) = \frac{1}{w \times h} \left( \frac{\sum_{i=1}^{w} \sum_{j=1}^{h} |A_{ij} - B_{ij}|}{2^L - 1} \right) \times 100\%,$$

where $L$, $A$, $B$, $w$, and $h$ have the same definition as with NPCR. According to Kwoka, and Tang’s report [17], for two random images,
The expected UACI value is

\[ \frac{\sum_{i=1}^{2^3-1} (i+1)}{2^1 \times 2^3 \times (2^3 - 1)} \times 100\%. \]

When secret images are binary and grayscale images, the \( L = 1 \) and the expected UACI value is

\[ \frac{\sum_{i=1}^{2^1-1} (i+1)}{2^1 \times 2^1 \times (2^1 - 1)} \times 100\% = \frac{2}{4} \times 100\% = 50\%. \]

When secret images are color images, three bits are required to represent one pixel of the shadow because each shadow contains three planes. Therefore, the expected UACI value is

\[ \frac{\sum_{i=1}^{2^3-1} (i+1)}{2^3 \times 2^3 \times (2^3 - 1)} \times 100\% = \frac{\sum_{i=1}^{7} (i+1)}{2^6 \times (2^3 - 1)} \times 100\% = \frac{168}{448} \times 100\% = 37.5\%. \]

Table 6 shows the experimental results of our scheme.

<table>
<thead>
<tr>
<th>Original image</th>
<th>Lena</th>
<th>Jet</th>
<th>Pepper</th>
<th>Sailboat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halftone logo</td>
<td><img src="halftone1" alt="Image" /></td>
<td><img src="halftone2" alt="Image" /></td>
<td><img src="halftone3" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Shadow 1</td>
<td><img src="shadow1" alt="Image" /></td>
<td><img src="shadow2" alt="Image" /></td>
<td><img src="shadow3" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Shadow 2</td>
<td><img src="shadow4" alt="Image" /></td>
<td><img src="shadow5" alt="Image" /></td>
<td><img src="shadow6" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original image</th>
<th>Sailboat</th>
<th>Tiffany</th>
<th>Pepper</th>
<th>Lena</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halftone logo</td>
<td><img src="halftone1_grayscale" alt="Image" /></td>
<td><img src="halftone2_grayscale" alt="Image" /></td>
<td><img src="halftone3_grayscale" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Shadow 1</td>
<td><img src="shadow1_grayscale" alt="Image" /></td>
<td><img src="shadow2_grayscale" alt="Image" /></td>
<td><img src="shadow3_grayscale" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>Shadow 2</td>
<td><img src="shadow4_grayscale" alt="Image" /></td>
<td><img src="shadow5_grayscale" alt="Image" /></td>
<td><img src="shadow6_grayscale" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>
Table 5
Set of shadows for the color test images.

![Original image](Barbara) ![Jet] ![Watch] ![Zelda]

![Halftone logo](logo)

![Shadow 1](shadow1)

![Shadow 2](shadow2)

Table 6
Values of NPCR and UACI for two shadows of test images.

<table>
<thead>
<tr>
<th>Binary images</th>
<th>NPCR (%)</th>
<th>UACI (%)</th>
<th>Grayscale images</th>
<th>NPCR (%)</th>
<th>UACI (%)</th>
<th>Color images</th>
<th>NPCR (%)</th>
<th>UACI (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>49.0</td>
<td>49.0</td>
<td>Sailboat</td>
<td>43.3</td>
<td>43.3</td>
<td>Barbara</td>
<td>70.1</td>
<td>32.8</td>
</tr>
<tr>
<td>Jet</td>
<td>47.8</td>
<td>47.8</td>
<td>Tiffany</td>
<td>47.4</td>
<td>47.4</td>
<td>Jet</td>
<td>77.9</td>
<td>33.5</td>
</tr>
<tr>
<td>Pepper</td>
<td>51.3</td>
<td>51.3</td>
<td>Pepper</td>
<td>48</td>
<td>48</td>
<td>Watch</td>
<td>69.8</td>
<td>34.7</td>
</tr>
<tr>
<td>Sailboat</td>
<td>47.4</td>
<td>47.4</td>
<td>Lena</td>
<td>56.2</td>
<td>56.2</td>
<td>Zelda</td>
<td>62.3</td>
<td>36.2</td>
</tr>
</tbody>
</table>

From experimental results listed in Table 6, we can see when one pixel is changed; not only the percentage of changed pixels in its corresponding shadow is significant but also the UACI is significant. With regard to obtained results listed in Table 6, it seems that the proposed scheme has a good ability to resist differential attack.

4.2. Accuracy

In our experiments, the peak signal-to-noise ratio (PSNR) defined in Eq. (10) is used to evaluate the quality of the reconstructed images (both grayscale and color images). A higher PSNR means that the quality of the reconstructed secret image is better. Typical PSNR value ranges from 20 to 40 dB.

\[
PSNR = 10 \times \log_{10} \frac{255^2}{MSE}.
\]  

where MSE is the mean square error between the original image and the reconstructed one. For an original grayscale image with a size of \(w \times h\), the formula for MSE is defined by using

\[
MSE_{\text{grayscale}} = \frac{1}{w \times h} \sum_{x=1}^{w} \sum_{y=1}^{h} (G_{xy} - C_{xy})^2,
\]  

where \(G_{xy}\) and \(C_{xy}\) are the pixel values at position \((x,y)\) of the original grayscale image and the reconstructed grayscale secret image, respectively. For an original color image \(w \times h\) in size, the MSE is defined using

\[
MSE_{\text{color}} = \frac{1}{w \times h} \sum_{x=1}^{w} \sum_{y=1}^{h} \left( \frac{(RG_{xy} - RG'_{xy})^2 + (GG_{xy} - GG'_{xy})^2 + (BG_{xy} - BG'_{xy})^2}{3} \right),
\]  

where \(RG_{xy}, RG'_{xy}, GG_{xy}, GG'_{xy}, BG_{xy},\) and \(BG'_{xy}\) are the pixel values at position \((x,y)\) of the Red, the Green, and the Blue planes in the original color image and the reconstructed color image, respectively.

Moreover, based on the procedures of the shares construction phase and revealing and verifying phase described in Section 3, we can see that the original image and the reconstructed image are the same with this scheme when the secret image is binary image. The experimental results given in Tables 7–9 demonstrate that the quality of the reconstructed image corresponds to the three test images sets shown in Figs. 12(1), (2), and (3).

Added to the fact that the reconstructed binary image is the same as the original image when the secret image is binary image; the average image quality of the reconstructed grayscale images and of the reconstructed color image is 33.44 and 33.71 dB, respectively. In other words, the reconstructed image is almost indistinguishable to the original image when subjected to human perception. Thus, our proposed scheme completely satisfies the accuracy requirement of VSS scheme.

Because an inverse halftone technique is adopted in the revealing and verifying phase, the image quality of both reconstructed
Table 7
Quality of the reconstructed image with binary test images.

<table>
<thead>
<tr>
<th>Original binary image</th>
<th>Reconstructed binary image</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Table 8
Quality of the reconstructed image with grayscale test images.

<table>
<thead>
<tr>
<th>Original grayscale image</th>
<th>Reconstructed grayscale image</th>
<th>Original halftone logo</th>
<th>Average PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td>33.44dB</td>
</tr>
<tr>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Average PSNR represents the average PSNR of the four reconstructed grayscale images.

grayscale and color images depends completely on the halftoning and inverse halftoning techniques. The PSNRs reported in Tables 8 and 9 are derived from the reconstructed secret images that have been generated by using Chung and Wu’s edge lookup inverse halftoning (ELIH) as the halftoning and inverse halftoning techniques for this scheme. To prove that Chang and Wu’s scheme can provide higher quality reconstructed images, Table 10 compares the PSNRs of the reconstructed grayscale images created by using Chung and Wu’s ELIH scheme and those created by using other inverse halftoning techniques. From Table 10, we can see Chung and Wu’s scheme provides higher image quality in the reconstructed secret image, which confirms that our selection is correct.

### 4.3. Computational complexity

For the two phases described in Sections 3.1 and 3.2, we can see that the computational cost of our scheme depends on two operations: halftoning and bitwise operations. Obviously, the complexity of the second operation is very low and does not have much effect on the computational complexity of our scheme. In fact, our scheme incurs no computational complexity when performed on binary images. Hence, the computational cost of the halftoning operation plays an important role in evaluation optimization when our scheme is performed on grayscale and color images. Two techniques, error diffusion and ELIH, were chosen to maintain the quality of the reconstructed image. Although these two techniques add some
Table 9
Quality of the reconstructed image with color test images.

<table>
<thead>
<tr>
<th>Original color image</th>
<th>Reconstructed color image</th>
<th>Original halftone logo</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Original image" /></td>
<td><img src="image2" alt="Reconstructed image" /></td>
<td><img src="image3" alt="Original halftone" /></td>
</tr>
<tr>
<td><img src="image4" alt="PSNR = 33.75 dB" /></td>
<td><img src="image5" alt="PSNR = 33.51 dB" /></td>
<td><img src="image6" alt="PSNR = 33.21 dB" /></td>
</tr>
<tr>
<td><img src="image7" alt="PSNR = 34.37 dB" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Average PSNR represents the average PSNR of the four reconstructed color images.

Table 10
Compare the quality of the reconstructed grayscale images.

<table>
<thead>
<tr>
<th>Images</th>
<th>Sailboat (dB)</th>
<th>Tiffany (dB)</th>
<th>Lena (dB)</th>
<th>Pepper (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chung and Wu's scheme [11]</td>
<td>33.84</td>
<td>32.63</td>
<td>33.59</td>
<td>33.69</td>
</tr>
<tr>
<td>Mese et al.'s scheme [19]</td>
<td>26.76</td>
<td>27.01</td>
<td>25.84</td>
<td>26.08</td>
</tr>
<tr>
<td>Neelamani et al.'s scheme [22]</td>
<td>25.76</td>
<td>29.46</td>
<td>25.01</td>
<td>25.51</td>
</tr>
</tbody>
</table>

Table 11
Computational complexity of our scheme with grayscale and color test images.

<table>
<thead>
<tr>
<th>Grayscale images</th>
<th>Color images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sailboat</td>
<td>Tiffany</td>
</tr>
<tr>
<td>09 m20 s</td>
<td>8 m36 s</td>
</tr>
</tbody>
</table>

computational complexity to our scheme, it is still acceptable. The experimental results, which offer the execution time of our scheme, are shown in Table 11. Because the lookup table can be generated in advance, the computational cost for generating the lookup table is not included here.

4.4. Pixel expansion

In our scheme, the correlation between the size of the original image and the shadow can be divided into two cases corresponding to the secret images, the binary and secret images which are the natural grayscale and color images. In the first case, the size of each shadow is the same as that of the original binary image. In the second case, when the secret images are natural image, including grayscale and color images, the size of each shadow is one-eighth of size of the original image. The experimental results, which demonstrate the correlation between the original image size and the shadow size, are shown in the Table 12.

4.5. Verifying evaluation

In our scheme, the mean square error (MSE) shown in Eqs. (11) and (12) is also used to compute the content of an original halftone logo with an extracted halftone logo to identify the cheating. If the MSE value is equal to zero, there is no difference between the two logos. No cheating is detected and the reconstructed grayscale image is totally believable. If the MSE value is not equal to zero, the participants come under suspicion. They can ask dealer to publish the original halftone logo and compare it with the extracted halftone logo. The experimental results were conducted with two assumptions corresponding to two circumstances. The first is that no cheating is caused by any participant or transmission. The reconstructed image quality and the extracted halftone logo, shown in Tables 13 and 14 to demonstrate the reliability of our scheme, correspond to a set of grayscale images and a set of color images. The second circumstance, shown in Table 15, assumes that some changes have occurred in the shadows set due to action by a dishonest participant.

To measure the reliability of the received set of shadows and the reconstructed secret image, the reliability parameter uses the values “Sure” or “Not sure”. If the MSE value of the original halftone logo and the extracted halftone logo is zero, the parameter will be “Sure”, and vice versa.

To illustrate verification ability of the proposed scheme, the shadows set is assumed to be altered by the dishonest participants. The quality of the reconstructed grayscale image and the reliability are presented in Table 15. Four scenarios of defrauding are considered as follows:

- “Sailboat” is the original secret image and CCU is the logo. The first shadow is replaced by a fake shadow, which is composed of halftone image “Pepper” and the second shadow of “Sailboat” under XOR operations.
- “Tiffany” is the original secret image and CCU is the logo. The first shadow is not changed but the second shadow is replaced by a fake shadow composed of halftone image “Pepper” and another image under XOR operations.
- “Pepper” is the original secret image and MSN is the logo. The first shadow is replaced by a fake shadow, which is the first shadow when “Tiffany” is the original secret image, and CCU is the logo.
- “Lena” is the original secret image and MSN is the logo. The first shadow is replaced by fake shadow, which is the second shadow when “Pepper” is the original secret image, and MSN is the logo.

From Table 15, we can see that if there is a lack of accuracy in the first or the second shadow, the reconstructed secret image cannot be exactly generated. After comparing the original halftone logo with the extracted logo, we see that no matter how doubtful the first shadow is, it cannot successfully generate the reconstructed secret image. However, our experiment also pointed out a weakness in our scheme. When the second shadow is faked and the
Table 12
Correlation between original image size and its shadow size.

<table>
<thead>
<tr>
<th>Original image vs. shadows</th>
<th>Binary image (128×128) (bits)</th>
<th>Grayscale image (128×128) (bytes)</th>
<th>Color image (128×128) (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of the original image</td>
<td>16,384</td>
<td>16,384</td>
<td>2048</td>
</tr>
<tr>
<td>Size of the shadow 1</td>
<td>16,384</td>
<td>2048</td>
<td>6144</td>
</tr>
<tr>
<td>Size of the shadow 2</td>
<td>16,384</td>
<td>2048</td>
<td>6144</td>
</tr>
</tbody>
</table>

Table 13
Image quality of the reconstructed grayscale, extracted halftone logo error and reliability conclusion when no cheating is detected.

<table>
<thead>
<tr>
<th>Original grayscale image</th>
<th>Reconstructed grayscale image</th>
<th>PSNR = 33.84 dB</th>
<th>PSNR = 32.63 dB</th>
<th>PSNR = 33.60 dB</th>
<th>PSNR = 33.70 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original halftone logo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extracted halftone logo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>Sure</td>
<td>Sure</td>
<td>Sure</td>
<td>Sure</td>
<td></td>
</tr>
</tbody>
</table>

Table 14
Image quality of the reconstructed color image quality, extracted halftone logo error and reliability conclusion when no cheating is detected.

<table>
<thead>
<tr>
<th>Original color image</th>
<th>Reconstructed color image</th>
<th>PSNR = 33.75 dB</th>
<th>PSNR = 33.51 dB</th>
<th>PSNR = 33.21 dB</th>
<th>PSNR = 34.37 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original halftone logo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extracted halftone logo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>Sure</td>
<td>Sure</td>
<td>Sure</td>
<td>Sure</td>
<td></td>
</tr>
</tbody>
</table>

reconstructed secret image is not correct, the halftone logo can still be correctly reconstructed. Although this exceptional case may occur, it is worth while to note that the reconstructed logo is still a noise-like image with our proposed scheme. The reconstructed noise-like logo provides another clue to assist honest participants in determining whether the reconstructed “Pepper” is a wrong result. Moreover, in this case, participants still can ask dealer to offer the original logo so that participants can compare the extracted logo and the original logo to obtain a sound conclusion. In other words, our cheating prevention function still works in such exceptional case.

4.6. Discussion and comparison

To further evaluate our scheme’s performance, this subsection compares our scheme with other VSS schemes [9,15,23,28,32,33] designed with a cheating prevention mechanism. To clearly
Table 15
Reconstructed grayscale image quality, extracted halftone logo and the extracted result when cheating occurs.

<table>
<thead>
<tr>
<th>Original grayscale image</th>
<th>Reconstructed grayscale image</th>
<th>Original halftone logo</th>
<th>Extracted halftone logo</th>
<th>Cheating type</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td>(1)</td>
<td>Not sure</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td>(2)</td>
<td>Sure</td>
</tr>
<tr>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
<td>(3)</td>
<td>Not sure</td>
</tr>
<tr>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
<td>(4)</td>
<td>Not sure</td>
</tr>
</tbody>
</table>

demonstrate the advantage of our scheme, the comparisons are divided into categories and listed in Tables 16 and 17, depending on whether the secret image is a binary image or the natural image. Seven evaluation criteria are listed in Tables 16 and 17 to facilitate comparing our scheme with other VSS schemes: number of shadows \((n)\), MSE value between the reconstructed secret image and the original secret image, shadow size, computational complexity, shadow style, additional information and cheating prevention mechanism. Fuller descriptions of the seven criteria follow:

1. **Number of shadows \((n)\):** The value of \(n\) is the number of shadows generated in the shares construction phase of a VSS scheme.
2. **MSE value between the reconstructed secret image and the original secret image:** This value is to evaluate the similarity between two images. If the MSE value is zero, the reconstructed secret image and the original secret image are the same, and vice versa.
3. **Shadow size:** This criterion considers the pixel expansion problem and compares the size of the original secret image with each shadow.
4. **Computational complexity:** This criterion is evaluated by the number of operations and time consumed.
5. **Shadow style:** This criterion considers the set of shadows to be the like-noise images or meaningful images.
6. **Additional information:** This criterion evaluates whether extra information is required to perform the cheating prevention function.
7. **Cheating prevention mechanism:** This criterion identifies the objective of the cheating prevention mechanism in \((2,n)\) VSS scheme [15,23,28] and ours.

Table 16 compares our scheme with other \((2,n)\) VSS schemes when the secret image is a binary image.

From Table 16, we can see that when our scheme is applied to binary secret images, the reconstructed secret image is exactly the same as the original secret image. With other schemes [15,23,28], which use a stacking operation to obtain the reconstructed secret image, the reconstructed images are not the same as the original secret image. While our scheme has a fixed shadow size that is the same as that of the original secret image, other schemes [15,23,28] has a tradeoff between the number of shadows \((n)\) and the shadow size. In general, the shadow size in other schemes is bigger than the original secret image even when the number of shadows is set as \(n = 2\). In other words, our scheme is the only scheme that does not experience pixel expansion. In computational complexity, Horng et al. [15], Prisco and Santis [23] and Tsai et al. [28] use a stacking operation to obtain the reconstructed secret image. During the shares construction phase, Horng et al. [15] and Prisco and Santis’s [23] schemes adopt a visual cryptography approach to generate shadows. Tsai et al.’s scheme applies a GA algorithm rather than visual cryptography to generate shadows during the shares construction phase [28]. In contrast, our scheme uses only a Boolean operation to generate \(n\) shares and to later obtain the reconstructed secret image. Moreover, the halftoning and inverse halftoning operations are not necessary when the secret image is binary image; therefore, our scheme is lower in computational complexity than [15,23,28]. In additional information, Horng et al.’s first scheme [15] called \((k,n)\), required extra information called a verification logo from each participant. Horng et al.’s second scheme [15], called \((k,n+l)\) VSS, divides the secret image into \(n+l\) shares instead of \(n\) shares. The set of \(l\) shares is considered to be additional information. Prisco and Santis’s scheme [23] uses the base matrices to overcome the requirement for additional information in Horng et al. [15]. Tsai et al.’s schemes [28] solve the cheating problem without extra burdens by adopting multiple distinct secret images. In our scheme, a hidden logo is extracted from the reconstructed secret image to verify the reconstructed secret image; therefore, no extra information is required. Thus, except for Horng et al.’s schemes [15], other similar schemes have no additional information problem. As for cheating prevention
mechanism, Horng et al. [15] help honest participants to verify whether their received shadows are correct by asking each participant to send a verification logo to the dealer at the beginning. Prisco and Santis [23] focus on the shadow design to prevent cheaters from generating fake shadows and forcing honest participants to reconstruct an incorrect secret image, rather than providing verification or authentication for received shadows or a reconstructed secret image. Tsai et al.’s scheme [28] uses the same strategy as Prisco and Santis except that it uses the generic algorithm instead of visual cryptography to be the kernel method. In Tsai et al.’s scheme, the dealer generates \(n/2! \times (n-2)!\) distinct homogeneous secret images from the original secret image. To create \(n\) shares, each pixel in a secret image is encoded independently. Therefore, each pixel in the same spatial location in all the distinct secret images is used to create a two-dimensional chromosome. In this way, cheaters have difficulty designing a valid fake shadow. In contrast with other schemes, we allow participants to authenticate the reconstructed secret image based on the extracted logo during the revealing phase.

Detailed descriptions of the seven evaluation criteria in Table 17 follow.

(1) **Number of shadows (n)**: Our scheme generates two shadows in the shares construction phase. However, other schemes can divide the secret image into an unlimited number of shadows.

(2) **MSE value between the reconstructed secret image and the original secret image**: Chang et al. [9] and Yang et al.’s [32] schemes extend the steganography mechanism in combination with the Chinese Remainder Theorem (CRT), by means of the Galois Field \(GF(2^8)\). Both schemes [9,32] divide the secret image into \(n\) shadows by a \((k-1)\)-degree polynomial with the value of the prime number set as 251. There is some information lost from the pixel having a value larger than 250. Our scheme uses a halftoning technique to transform a grayscale or color image into a binary image during the shares construction phase. Therefore, some distortion occurs in the reconstructed secret image. In contrast, Zhao et al.’s scheme [33] uses the lossless image secret sharing method to handle any pixel values that are larger than 250 during the shares construction phase; therefore, their reconstructed secret image is the same as the original secret image.

(3) **Shadow style**: Both schemes in [9,32] embed the shadow into stego-images by using a steganography mechanism. The stego-image received by the participant is larger than the original secret image. Although the size of each shadow in Zhao et al.’s scheme [33] is calculated as \(1/t\) of the original image and \(t\) is chosen as 2, as usual, our scheme still outperforms Zhao et al.’s scheme because we can provide the smallest shadow size, which is equal to one-eighth of the original secret image.

(4) **Computational complexity**: The three VSS schemes in [9,32,33] employ \((k-1)\)-degree polynomial to divide the secret data into \(n\) shares. Moreover, to perform authentication, Chang et al.’s scheme [9] uses the CRT in combination with a steganography mechanism, and Yang et al.’s scheme [32] uses the Galois Field \(GF(2^8)\) in combination with a steganography mechanism. Zhao et al. [33] use the intractability of the discrete logarithm to deal with the authentication problem. Meanwhile, our scheme uses only a Boolean operation to perform during both the shares construction phase and the verifying and revealing phase. Moreover, the lookup table for inverse halftoning can be generated in advance. Thus, when compared with other schemes [9,32,33], our scheme is relatively much less complex.

(5) **Cheating prevention function**: Chang et al. and Yang et al. [9,32] employ the steganography mechanism to hide generated shadows into the stego-image before sending the shadows to participants. Thus, the set of shadows held by participants is made up of

### Table 16
Comparison of our scheme with the other (2, \(n\)) VSS schemes when the secret image is a binary image.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of shadows ((n))</td>
<td>(n \geq 2)</td>
<td>(n \geq 2)</td>
<td>(n = 2)</td>
</tr>
<tr>
<td>MSE of the reconstructed image and the original image</td>
<td>MSE &gt; 0</td>
<td>MSE &gt; 0</td>
<td>MSE = 0</td>
</tr>
<tr>
<td>Shadow style</td>
<td>Noise-like shadow</td>
<td>Noise-like shadow</td>
<td>Noise-like shadow</td>
</tr>
<tr>
<td>Computational complexity</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Cheating prevention function</td>
<td>Uses own verification logo</td>
<td>Uses base matrices to make it difficult for cheaters to design valid fake shadows</td>
<td>Uses extracted logo to verify the reconstructed secret image</td>
</tr>
</tbody>
</table>

### Table 17
Comparison of our scheme and the other (2, \(n\)) VSS schemes when the secret image is a natural image.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of shadows ((n))</td>
<td>(n \geq 2)</td>
<td>(n \geq 2)</td>
<td>(n = 2)</td>
</tr>
<tr>
<td>MSE of the reconstructed image and the original image</td>
<td>MSE &gt; 0</td>
<td>MSE &gt; 0</td>
<td>MSE = 0</td>
</tr>
<tr>
<td>Shadow size</td>
<td>N=2</td>
<td>N=1.15</td>
<td>N/8</td>
</tr>
<tr>
<td>Shadow style</td>
<td>Meaningful shadow</td>
<td>Meaningful shadow</td>
<td>Noise-like shadow</td>
</tr>
<tr>
<td>Computational complexity</td>
<td>High</td>
<td>High</td>
<td>Relative low</td>
</tr>
<tr>
<td>Cheating prevention function</td>
<td>Uses stego-image with the embedded authentication code to authenticate shadow</td>
<td>Uses stego-image with the embedded authentication code to authenticate shadow</td>
<td>Uses some published information to identify cheaters</td>
</tr>
</tbody>
</table>

- **Table 16**: Comparison of our scheme with the other (2, \(n\)) VSS schemes when the secret image is a binary image.
- **Table 17**: Comparison of our scheme and the other (2, \(n\)) VSS schemes when the secret image is a natural image.
meaningful images that leak no information about the secret image. Experimental results show that both our scheme and Zhao et al.’s scheme [33] generate a set of noise-like images as shadows. Note that experimental results discussed in Section 4.2 confirm that our shadows leak no information about the secret image and can withstand a differential attack.

(6) Additional information: Both schemes in [9,32] embed the authentication code into the cover image along with the generated shadows. Since the information required to generate the reconstructed secret image and to authenticate the secret image is carried by the stego-image, no additional information is required to send to participants or to be published in advance. In contrast, Zhao et al.’s scheme [33] requires the use of some published information to authenticate the shadow. No additional information is required in our scheme. Therefore, except for Zhao et al.’s scheme [33], the remaining schemes [9,32] and our proposed scheme need no additional information.

(7) Cheating prevention function: Note that the schemes of Chang et al. [9] and Yang et al. [32] embed some authentication codes along with the generated shadows into the cover image at the shares construction phase. Later, these hidden authentication codes are used to authenticate the integrity of the stego-image. Zhao et al. [33] focus on identifying cheaters by using some published information, while ours aims to assist honest participants to verify the reconstructed secret image by using the extracted halftone logo.

5. Conclusions

In this paper, we propose a novel (2, 2) verifiable secret sharing scheme for both binary and natural images. Although in our scheme the number of shadows is limited to 2, it can work on binary, grayscale and color images without experiencing pixel expansion. Moreover, it provides verification function of reconstructed secret images by using an embedded halftone logo without having to send extra information through another channel. Error diffusion, halftoning transform, and image clustering are three techniques employed as foundation of our scheme. Based on the Boolean operator XOR, the proposed scheme can easily recover the reconstructed image from the collected shadows during the revealing and verifying phase. Because the lookup table for inverse halftoning can be performed in advance, and the second shadow can be easily generated by performing the Boolean operator XOR on the first shadow and the permuted halftone secret image, the computational cost of the shares construction remains relatively low even when the secret image is either grayscale or color image.

Experimental results confirm that our proposed scheme not only gives high reconstructed image quality with a PSNR ranging from 32.63 to 34.37 dB regardless of whether the images are grayscale or color. For the grayscale and color images, our scheme offers shadows with only one-eighth the size of the original secret image and reconstructs the secret image with slight distortion. When the secret image is a binary image, our scheme not only generates shadows with the same size of the secret image but also reconstructs the secret image without causing any distortion. Based on condition of the extracted halftone logo and the MSE between the extracted halftone logo and the original halftone logo, the proposed scheme allows honest participants to verify the reliability of the reconstructed secret image. If the MSE value is equal to zero, there is no cheating by a dealer or participants. When the MSE value is not equal to zero, even if the reconstructed image is meaningful the reconstructed image can be determined to be incorrect. By using our proposed scheme, participants do not need to perform the verification procedure on shadows received from a dealer and from other participants. They only need to perform verification once to determine whether a reconstructed secret image is correct. Therefore, our proposed scheme’s verification procedure is more efficient than with other similar schemes.

References


About the Author—CHIN-CHEN CHANG received his B.S. degree in Applied Mathematics in 1977 and his M.S. degree in Computer and Decision Sciences in 1979 from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D. in Computer Engineering in 1982 from National Chiao Tung University, Hsinchu, Taiwan. From 1983 to 1989, he was the Faculty at the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. Since August 1989, he was a Professor of the Institute of Computer Science and Information Engineering at National Chung Cheng University, Chia-yi, Taiwan. Dr. Chang is a Fellow of IEEE and a Fellow of IEE. He is also a member of the Chinese Language Computer Society, the Chinese Institute of Engineers of the Republic of China, and the Phi Tau Phi Society of the Republic of China. His research interests include database design, computer cryptography, and data compression.

About the Author—CHIA-CHEN LIN received her B.S. degree in Information Management in 1992 from the Tamkang University, Taipei, Taiwan. She received both her M.S. degree in Information Management in 1994 and Ph.D. degree in Information Management in 1998 from the National Chiao Tung University, Hsinchu, Taiwan. Dr. Lin is currently a Professor of the Department of Computer Science and Information Management, Providence University, Sha-Lu, Taiwan. Since August 2008, she is the Associate Dean of Academic Affairs of Providence University. She is also a member of IEEE and ACM. Her research interests include image and signal processing, image hiding, mobile agent, and electronic commerce.

About the Author—T. HOANG-NGAN LE received her B.S. degree in Information Technology in 2005. From Sep. 2006 to Jan. 2008, she had been a Faculty Member as well as a Master Student of Faculty of Information Technology, University of Science, Vietnam National University-HCMC, Vietnam. From Jan. 2008 to Jan. 2009, she had studied and researched in MSN lab, Feng Chia University, Taichung, Taiwan. She received her M.S. degree in Information Technology in Apr. 2009 from the same university. She is currently a Lecturer of Department of Computer Science, Faculty of Information Technology, University of Science, Vietnam National University-HCMC, Vietnam. Her current research interests include image processing, audio processing, and multimedia security.

About the Author—HOAI BAC LE received his B.S. degree in Mathematics in 1984 from University of Education, Vietnam and the M.S. degree in Computer Sciences in 1990 from University of Education, Vietnam. He received his Ph.D. in Mathematics for Computers and Computing Systems in 2000 from the Natural Sciences University. Right now, he is the Head of Computer Science Division as well as the Vice Dean, in charge of Postgraduate Training. His current research interests include artificial intelligence, soft computing, knowledge discovery and data mining and data hiding.