Data Hiding in Halftone Images by XOR Block-Wise Operation with Difference Minimization

Ching-Nung Yang[1], Guo-Cin Ye[1], and Cheonshik Kim[2]
[1] Dept. of Computer Science and Information Engineering, National Dong Hwa University,
#1, Sec. 2, Da Hsueh Rd., Hualien, Taiwan
[e-mail: cnyang@mail.ndhu.edu.tw, m9624015@ems.ndhu.edu.tw]
[2] Department of Computer Engineering, Sejong University
98 Gunja-dong, Gwangjin-gu, Seoul 143-747, Korea
[e-mail: mipsan@paran.com, mipsan@sejong.ac.kr]
*Corresponding author: Cheonshik Kim

Received September 25, 2010; revised December 30, 2010; accepted January 20, 2011; published February 28, 2011

Abstract

This paper presents an improved XOR-based Data Hiding Scheme (XDHS) to hide a halftone image in more than two halftone stego images. The *hamming weight* and *hamming distance* is a very important parameter affecting the quality of a halftone image. For this reason, we proposed a method that involves minimizing the *hamming weights* and *hamming distances* between the stego image and cover image in 2×2-pixel grids. Moreover, our XDHS adopts a block-wise operation to improve the quality of a halftone image and stego images. Furthermore, our scheme improves security by using a block-wise operation with A-patterns and B-patterns. Our XDHS method achieves a high quality with good security compared to the prior arts. An experiment verified the superiority of our XDHS compared with previous methods.

Keywords: Data hiding, halftone image, XOR operation, visual cryptography

DOI: 10.3837/tiis.2011.02.013

1. Introduction

A halftone image is composed of 0 and 255-bit pixels, and the series of pixel patterns creates the illusion of a multi-tone image when viewed from a distance by the human visual system (HVS). Such halftone images are required for some specific applications, e.g., scanned text, figures, signatures, books, newspapers, magazines, or digital printers, which cannot print continuous tones. Moreover, halftone images have been applied in everyday life, e.g., a scanned handwritten signature captured by a PDA is typically used to pay for various services. The increasing demand for halftone images has motivated researchers to study methods of embedding data and watermarks into them. Thus, various data hiding and watermarking techniques for halftone images have been proposed [1][2][3][4][5][6][7][8][9][10]. In reference [1], a secret image was embedded into *k* halftone images by a simple XOR operation in bit-planes. In [2][3], the authors adopted an error diffusion dithering technique to hide a secret image that would appear when the halftone images were overlaid. Some schemes have manipulated "flippable" pixels to embed a significant amount of data without causing noticeable vestiges [4][5][6][7]. The schemes in [8][9] used block patterns to represent the secret data. In [10], a pair-wise logical computation was used to design a reversible data hiding scheme that could achieve the lossless reconstruction of a halftone image. With the exception of the schemes in [2][3], none of the above-mentioned schemes provide the stacking-to-see property in decoding. The data hiding schemes in [2][3] could stack (OR-ed operation) the stego-images (modified halftone images) to reveal the secret visually by HVS directly. This distinguishing property (stacking-to-see) of decoding can be used to securely and cheaply share the secret information, i.e., an OR-based DH scheme (ODHS). While [2][3] presented easy decoding schemes, the reconstructed secret had low contrast. However, the schemes in [2][3] are unsuitable for hiding a natural halftone image (note: the term natural halftone image comes from [11] and refers to a photographed image that is converted to a binary image and can be observed similar to the original image by HVS). Therefore, we propose an improved XOR-based data hiding scheme (XDHS) to considerably enhance the visual quality of stego-images and the reconstructed image. Moreover, this scheme makes it possible to hide a natural halftone image. The rest of this paper is organized as follows. Section 2 describes the related works and the design concept. The XDHS algorithm is presented in Section 3. Section 4 discusses the experimental work and security analysis, and finally Section 5 concludes the paper.

2. Related Works and Proposed Design Concept

2.1 Related Works

An ODHS [2][3] can also be implemented by a well-known visual cryptography scheme (VCS). A (k, n)-threshold VCS encrypts a secret image into n shadow images (shadows) by expanding a secret pixel into m sub-pixels. Any k ($k \le n$) shadows can be stacked (OR operation) to visually decode the secret image by HVS, but k–1 or fewer shadows will not show any information. The first VCS encrypted a black/white secret image into noise-like shadows [14]. Noise-like shadows are viewed with suspicion by censors and are difficult to identify and manage when delivered by e-mail or fax. Therefore, extended VCSs (EVCSs) with meaningful cover images (often natural images) on shadows were given in [15][16] to address the problems of suspicion and management. Recently, XOR-based EVCSs were proposed [17][18] to enhance the contrast of the reconstructed image. Obviously, the k stego-images ODHS (k-OHDS) and proposed k stego-images XDHS (k-XDHS) can be implemented by the OR-based (k, k)-EVCS and XOR-based (k, k)-EVCS, respectively. However, both EVCSs use expanded stego-images, which results in poor visual quality for the stego-images and reconstructed image. In addition, the XOR-based EVCS leaves cover image remnants on the reconstructed image. Such disadvantages make the XOR-based (k, k)-EVCS inappropriate for implementing our k-XDHS. In the proposed k-XDHS, there are (k+1) halftone images: k cover images ($I_1, I_2, \ldots, I_k$) and one secret image (I_{k+1}). We want to embed a modified secret image, I'_{k+1}, into k modified stego-images $(I'_1, I'_2, \ldots, I'_k)$. Image I'_{k+1} can be decoded by $I'_{k+1} = I'_1 \oplus I'_2 \oplus \ldots \oplus I'_k$. Our aim is to minimize the visual distortion between I_j and I'_j, $j \in [1, k+1]$. Obviously, a k-XDHS can be reduced to a (k–r)-XDHS by making any r images invariant and only modifying the pixels in the other (k+1– r) images. A reasonable application scenario is the recovery of a distortion-less halftone secret image by keeping secret image I_{k+1} invariant and modifying the other k cover images (I_1, I_2, …, I_k). References [1][4][5][6][7][8][9][10] provide data hiding schemes for halftone or binary images. These articles do not describe the relationship between pixels and image quality. On the other hand, we decribe the relationship between a block-wise operation and the quality of an image. The XOR operation in XDHS is a reversing-like operation, which was proved in [11][12][13]. In this paper, we will show the encoding and decoding algorithms used to get good visual quality in (k+1) modified halftone images.

2.2 Design Concept

A halftone image could be reproduced in gray scale by arranging the black and white pixels in a grid. HVS could average the region around a pixel instead of decoding every pixel individually, making it possible to create the illusion of many gray levels in a halftone image. Therefore, it is possible to make the black pixels in a grid simulate the shades of gray in an image. Thus, it is natural that the more black pixels there are in a grid, the darker the grid will appear to be. **Fig. 1** shows five different types (G_0, G_1, G_2, G_3, G_4) of 2×2-pixel grids with 0, 1, 2, 3, and 4 black pixels representing five intensity levels. G_2 has six possible combinations, G_1 and G_3 have four combinations, and G_0 and G_4 only have one combination. Based on the observation that the same *Hamming weight* simulates an approximate gray level, we propose a block-wise operation (a 2 × 2-pixel grid) to design the k-XDHS. This block-wise operation minimizes the *Hamming weight* and *Hamming distance* in a 2×2-pixel grid. Suppose that $P_{i,1}$, $P_{i,2}$, …, $P_{i,k}$ represent the patterns of the ith block (a 2×2-pixel grid) in k cover images, and that $P_{i,k+1}$ is the pattern of the ith block in a secret image. The $P'_{i,j}$ patterns are the modified patterns of $P_{i,j}$, $j \in [1, k+1]$. Let $w(\cdot)$ and $d(\cdot)$ be the *Hamming weight* and *Hamming distance* functions, respectively. To obtain better visual quality in a halftone image, the modified patterns, $P'_{i,j}$, in the proposed k-XDHS should satisfy the following three conditions.

(X-1) $P'_{i,1} \oplus P'_{i,2} \oplus \cdots P'_{i,k} = P'_{i,k+1}$.

(X-2) Make $\Delta = \sum_{j=1}^{k+1} \Delta_j$ as small as possible, with $\sum_{j=1}^{k+1} \left| \Delta / k - \Delta_j \right|$ being the minimum,

where $\Delta_j = \left| w(P_{i,j}) - w(P'_{i,j}) \right|$, $1 \le j \le k+1$.

(X-3) Make $d = \sum_{j=1}^{k+1} d_j$ as small as possible, with $\sum_{j=1}^{k+1} \left| d/k - d_j \right|$ being the minimum,

where $d_j = d(P_{i,j} - P'_{i,j})$, $1 \le j \le k+1$.

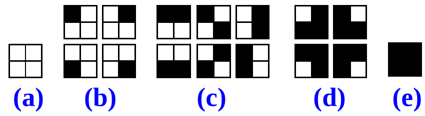


Fig. 1. Five 2×2-pixel grids and their combinations: **(a)** G_0 with 0 black pixels, **(b)** G_1 with 1 black pixel, **(c)** G_2 with 2 black pixels, **(d)** G_3 with 3 black pixels, **(e)** G_4 with 4 black pixels.

(**X-1**) is a decoding criterion and ensures a successful reconstruction of k-XDHS. (**X-2**) ensures that the ith blocks in all $(k+1)$ images have, on average, intensity levels close to their original patterns (*note*: the same *Hamming weight* has the same intensity). In (**X-3**), we arrange the black and white pixels in a 2×2-pixel grid according to the original one. This ensure that the halftone image is shaded appropriately and retains the contours. (**X-1**) can be referred to as the decoding criterion, and the latter two criteria are contrast conditions. Concerning the contrast criteria, we first make sure of satisfying (**X-2**), and then (**X-3**). We attempt to find a minimum Δ and Δ_j, $1 \le j \le k+1$, which are as similar as possible. Our aim is to keep the same *Hamming weight* in a pattern (the same number of black pixels in a grid), which ensures that a grid has the same intensity. After satisfying (**X-2**), we permute the black pixels in a grid to find the minimum *Hamming difference* for retaining the contours.

3. Proposed XDHS

A trivial construction is the *randomized* k-XDHS. Suppose that p_j is a pixel in I_j, $j \in [1, k+1]$. If p_{k+1} equals $p_1 \oplus p_2 \oplus \ldots \oplus p_k$ as it happens, we do nothing. When $p_{k+1} \neq p_1 \oplus p_2 \oplus \ldots \oplus p_k$, we apparently can change any one pixel, p_j, $j \in [1, k+1]$, to obtain a successful decoding. In general, we averagely distribute the modified pixels in these $(k+1)$ images to retain the same visual quality in all of the stego-images and the reconstructed image. Let $(p_{i,1}, p_{i,2}, \ldots, p_{i,k})$ and $(p'_{i,1}, p'_{i,2}, \ldots, p'_{i,k+1})$ be the i-pixels in $k+1$ original halftone images $(I_1, I_2, \ldots, I_k)$ and the modified i-pixels in the $k+1$ modified halftone images $(I'_1, I'_2, \ldots, I'_{k+1})$, respectively, where $i \in [1, (x \times y)]$ and the image size is $(x \times y)$. The formal encoding algorithm for the *randomized* k-XDHS is given as follows.

Algorithm 1: Encryption of the *randomized* k-XDHS

Input: k cover images I_1–I_k; one secret image I_{k+1}.

Output: $I'_1 \sim I'_k$. /* k stego-images */

1) Obtain the ith pixel $(p_{i,1}, p_{i,2}, \ldots, p_{i,k+1})$ from $(\boldsymbol{I_1, I_2, \ldots, I_{k+1}})$;

2) For i=1 to $(x \times y)$ do {

2-1) If $p_{i,1} \oplus p_{i,2} \oplus \ldots \oplus p_{i,k} = p_{i,k+1}$ then $(p'_{i,1}, p'_{i,2}, \ldots, p'_{i,k+1}) = (p_{i,1}, p_{i,2}, \ldots, p_{i,k+1})$ else *randomly* flips one pixel in these $k+1$ pixels to gain new $(p'_{i,1}, p'_{i,2}, \ldots, p'_{i,k+1})$.

/* this modification holds $p'_{i,1} \oplus p'_{i,2} \oplus \ldots \oplus p'_{i,k} = p'_{i,k+1}$ */

2-2) Put the pixels $(p'_{i,1}, p'_{i,2}, \ldots, p'_{i,k})$ back to $(I'_1, I'_2, \ldots, I'_k)$;};

3) Output k stego-images $(I'_1, I'_2, \ldots, I'_k)$.

In the proposed k-XDHS, we use a block-wise operation (a 2×2-pixel grid) instead of a bit-wise operation in the *randomized* k-XDHS. When satisfying the contrast conditions, (**X-2**) and (**X-3**), it is correct that even though the block-wise operation has more modified pixels than the bit-wise operation, it will minimize the visual distortion. We first describe our k-XDHS with k=2, and then extend the construction method from k=2 to k>2.

3.1 Proposed *k*-XDHS with *k* = 2

We describe the encrypting algorithm of our 2-XDHS using two cover images (I_1 and I_2) and one secret image (I_3). Suppose a 2×2-pixel grid of the i^{th} block in (I_1, I_2, I_3) is the pattern P=($P_{i,1}$, $P_{i,2}$, $P_{i,3}$), and the modified pattern in (I'_1, I'_2, I'_3) is P'=($P'_{i,1}$, $P'_{i,2}$, $P'_{i,3}$), where, $i\in[1, (x\times y)/4]$. Let a 3-tuple, H=(H_1, H_2, H_3), be the *Hamming weights* of pattern P, where H_1, H_2, $H_3\in\{0, 1, 2, 3, 4\}$. There are 35 combinations of H when we do not consider the order (H_1, H_2, H_3). (Note: $35=\binom{5}{1}+\binom{5}{2}\times 2+\binom{5}{3}$; for example, $\binom{5}{1}$ implies that H_1, H_2, and H_3 have the same *Hamming weight*, there are $\binom{5}{1}$=5 patterns, H=(0, 0, 0), (1, 1, 1), (2, 2, 2), (3, 3, 3), and (4, 4, 4), respectively.) All 35 patterns are shown in **Table 1** and **Table 2**. The 11 patterns (B1-B11) in **Table 1** satisfy $P_{i,1}\oplus P_{i,2}=P_{i,3}$ (condition (**X-1**)), while the 24 patterns (A1-A24) in **Table 2** do not satisfy condition (**X-1**). We call the B-patterns (B1-B11) the unchangeable patterns, where we can permute the pixels in P to satisfy condition (**X-1**) and the values of (H_1, H_2, H_3) do not require modification. The ($P_{i,1}$, $P_{i,2}$, $P_{i,3}$) in the A-patterns (A1-A24) do not satisfy condition (**X-1**) even though we permute the pixels.

Table 1. Eleven unchangeable B-patterns

Hamming weight in ($P_{i,1}$, $P_{i,2}$, $P_{i,3}$): (**H**$_1$, **H**$_2$, **H**$_3$)			
(0, 0, 0)=B1	(0, 3, 3)=B4	(1, 2, 3)=B7	(2, 2, 2)=B10
(0, 1, 1)=B2	(0, 4, 4)=B5	(1, 3, 4)=B8	(2, 3, 3)=B11
(0, 2, 2)=B3	(1, 1, 2)=B6	(2, 2, 4)=B9	

However, we may modify H=(H_1, H_2, H_3) to $H'=(H'_1, H'_2, H'_3)$ to change the A-patterns into B-patterns such that the patterns in H and H' satisfy condition (X-2). For example, H=(1, 3, 3) in A16 does not satisfy condition (**X-1**). By changing it to B4, B7, B8, and B11, these four patterns have the minimum Δ=1 and $\sum_{j=1}^{3}|\Delta/k-\Delta_j|$=4/3 when compared with A16. Consider another case: H=(3, 4, 4) in A23. There are two B-patterns, B11=(2, 3, 3) and B8=(1, 3, 4), where Δ=3. At this time, B11 has Δ_1=1, Δ_2=1, Δ_3=1, and $\sum_{j=1}^{3}|\Delta/k-\Delta_j|$=0, while B8 has Δ_1=2, Δ_2=1, Δ_3=0, and $\sum_{j=1}^{3}|\Delta/k-\Delta_j|$=2. Hence, A23 is modified to B11. In the proposed k-XDHS, we change A-patterns into B-patterns, which is why we call the A-patterns changeable patterns.

Notation Used

I_j	I_j of size (x×y), j∈[1, k+1]; two halftone cover images I_1, I_2, and one secret image I_3 for k=2
I'_j	I'_j of size ($x\times y$), $j\in[1, k+1]$; two stego-images I'_1, I'_2, and the reconstructed image I'_3 for k=2
($P_{i,1}$, $P_{i,2}$, $P_{i,3}$)	the 2×2-pixel grid of ith block in (I_1, I_2, I_3), $i\in[1, (x\times y)/4]$

$(P'_{i,1}, P'_{i,2}, P'_{i,3})$	the pattern of a 2×2-pixel grid in (I'_1, I'_2, I'_3)
B1-B11	the unchangeable patterns $(P_{i,1}, P_{i,2}, P_{i,3})$, shown in **Table 1**
A1-A24	the changeable patterns $(P_{i,1}, P_{i,2}, P_{i,3})$, shown in **Table 2**
$\mathbf{M}(\cdot)$	modify the patterns in [A1-A24] to patterns in [B1-B11] according to **Table 2**
$\mathbf{P}(\cdot)$	permute 4 pixels in $(P_{i,1}, P_{i,2}, P_{i,3})$ to $(P'_{i,1}, P'_{i,2}, P'_{i,3})$ where $P'_{i,1} \oplus P'_{i,2} = P'_{i,3}$; all permutations labeled as $\mathbf{P}(P_{i,1}, P_{i,2}, P_{i,3})$ are shown in Appendix **Table A-1**
r_i	the probability of the modified pixels in I'_j, $j\in[1, 3]$, e.g., $r_1=r_2=r_3=1/3$ implies that the number of modified pixels in I'_1, I'_2, and I'_3 are almost same
$\mathbf{PA}(\cdot)$	$\mathbf{PA}(I_1, I_2, \ldots, I_{k+1})=\{S_1, S_1, S_3\}$ is to partition $(k+1)$ images $(I_1, I_2, \ldots, I_{k+1})$ into three sets $\{S_1, S_1, S_3\}$, where $\lvert S_i\rvert\geq 1$ and $\lvert S_1\rvert+\lvert S_2\rvert+\lvert S_3\rvert=(k+1)$; without loss of generality we could partition $(I_1, I_2, \ldots, I_{k+1})$ into $S_1=\{I_1,\ldots,I_{\lvert S_1\rvert}\}$, $S_2=\{I_{\lvert S_1\rvert+1},\ldots,I_{\lvert S_2\rvert+\lvert S_1\rvert}\}$, and $S_3=\{I_{\lvert S_1\rvert+\lvert S_2\rvert+1},\ldots,I_{k+1}\}$.

The corresponding modified B-patterns for an A-pattern that satisfies condition (**X-2**) are shown in **Table 2**. Consequently, we describe the encrypting algorithm of our (2, 2)-XISSS. Some notations are defined above.

Table 2. Twenty-four changeable A-patterns and their modified B-patterns satisfying condition (X-2).

(H_1, H_2, H_3)	(H'_1, H'_2, H'_3)	Δ	(H_1, H_2, H_3)	(H'_1, H'_2, H'_3)	Δ	(H_1, H_2, H_3)	(H'_1, H'_2, H'_3)	Δ
(0, 0, 1)=A1	B1; B2	1	(0, 2, 4)=A9	B4; B7; B8	2	(1, 4, 4)=A17	B5; B8	1
(0, 0, 2)=A2	B2; B6	2	(0, 3, 4)=A10	B4; B5; B8	1	(2, 2, 3)=A18	B7; B9; B10; B11	1
(0, 0, 3)=A3	B6	3	(1, 1, 1)=A11	B2; B6	1	(2, 3, 4)=A19	B8; B9; B11	1
(0, 0, 4)=A4	B6	4	(1, 1, 3)=A12	B6; B7	1	(2, 4, 4)=A20	B8; B11	2
(0, 1, 2)=A5	B2; B3; B6	1	(1, 1, 4)=A13	B7; B9	2	(3, 3, 3)=A21	B11	1
(0, 1, 3)=A6	B7	2	(1, 2, 2)=A14	B3; B6; B7; B10	1	(3, 3, 4)=A22	B9; B11	2
(0, 1, 4)=A7	B7	3	(1, 2, 4)=A15	B7; B8; B9	1	(3, 4, 4)=A23	B11	3
(0, 2, 3)=A8	B3; B4; B7	1	(1, 3, 3)=A16	B4; B7; B8; B11	1	(4, 4, 4)=A24	B9	4

Algorithm 2: Encryption of the proposed 2-XDHS

Input: Two halftone cover images, I_1 and I_2; one halftone secret image, I_3.

Output: $E_{2,2}(I_1, I_2, I_3)=I'_1$ and I'_2. /* two stego-images */

1) Obtain the ith block $(P_{i,1}, P_{i,2}, P_{i,3})$ from (I_1, I_2, I_3);

2) For i=1 to $(x\times y)$ /4 do {

2-1) If $(P_{i,1}, P_{i,2}, P_{i,3})\in$[B1-B11] then go to step (2-2) else $(P_{i,1}, P_{i,2}, P_{i,3})=\mathbf{M}(P_{i,1}, P_{i,2}, P_{i,3})$;

/* this makes $\Delta=\Delta_1+\Delta_2+\Delta_3$ as small as possible with $\sum_{j=1}^{3}\lvert \Delta/3-\Delta_j\rvert$ being minimum, i.e., satisfies Condition (X-2) */

2-2) Obtain all $(P'_{i,1}, P'_{i,2}, P'_{i,3})$ by $\mathbf{P}(P_{i,1}, P_{i,2}, P_{i,3})$;

/* the permutation let $(P'_{i,1} \oplus P'_{i,2} = P'_{i,3})$ satisfying Condition (X-1) */

2-3) Find a pattern $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$ having the smallest $d = d_1 + d_2 + d_3$ with $\sum_{j=1}^{3}\left|d/3 - d_j\right|$ being minimum (note: if more than one pattern, choose a $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$ with r_1=r_2=r_3=1/3);
/* this let $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$ hold condition (X-3), and the modifications are averagely distributed in I'_1, I'_2 and I'_3 */

2-4) Put $\left(P'_{i,1}, P'_{i,2}\right)$ back to I'_1 and I'_2; };

3) Output two stego-images $E_{2,2}(I_1, I_2, I_3)$=(I'_1 , I'_2).

Two simple examples are given to easily understand the encryption of our 2-XDHS. Let P=(P_1, P_2, P_3) be a pattern in (I_1, I_2, I_3). **Example 1** shows how to process a B-pattern, while **Example 2** deals with an A-pattern.

Example 1: Encrypt a pattern P=(P_1, P_2, P_3)=$\left(\begin{smallmatrix}\blacksquare&\blacksquare\\\square&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\square\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\square&\square\end{smallmatrix}\right)$ into $P' = \left(P'_1, P'_2, P'_3\right)$ using the proposed 2-XDHS.

Because P is an unchangeable pattern, B11, and also $P_1 \oplus P_2$=$\left(\begin{smallmatrix}\blacksquare&\blacksquare\\\square&\blacksquare\end{smallmatrix}\right) \oplus \left(\begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\square\end{smallmatrix}\right) = \left(\begin{smallmatrix}\square&\square\\\blacksquare&\blacksquare\end{smallmatrix}\right) \neq P_3$, we can find all of the permutations of P satisfying $P_1 \oplus P_2$=P_3. These consist of the following six patterns, $P' = \left(P'_1, P'_2, P'_3\right)$ having the smallest d=4 with $\sum_{j=1}^{3}\left|d/3 - d_j\right|$ =8/3 being the minimum: $\left(\begin{smallmatrix}\blacksquare&\blacksquare\\\square&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\square&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\square\\\blacksquare&\square\end{smallmatrix}\right)$ with d_1=0, d_2=2, d_3=2; $\left(\begin{smallmatrix}\blacksquare&\square\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\square&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\square&\square\end{smallmatrix}\right)$ with d_1=2, d_2=2, d_3=0; $\left(\begin{smallmatrix}\square&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\square\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\square&\square\end{smallmatrix}\right)$ with d_1=2, d_2=2, d_3=0; $\left(\begin{smallmatrix}\blacksquare&\blacksquare\\\square&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\square\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\square&\blacksquare\\\blacksquare&\square\end{smallmatrix}\right)$ with d_1=0, d_2=2, d_3=2; $\left(\begin{smallmatrix}\square&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\square\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\square\\\square&\blacksquare\end{smallmatrix}\right)$ with d_1=2, d_2=0, d_3=2; $\left(\begin{smallmatrix}\blacksquare&\square\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\square\end{smallmatrix}\ \begin{smallmatrix}\square&\blacksquare\\\square&\blacksquare\end{smallmatrix}\right)$ with d_1=2, d_2=0, d_3=2. Choose a pattern $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$ and make the number of modified pixels in I'_1, I'_2, and I'_3 as similar as possible, i.e., r_1=r_2=r_3=1/3. All six patterns, P', have the same (H_1, H_2, H_3)=(3, 3, 2) as P; thus, they have similar intensities. Moreover, the minimum difference in the *Hamming distance* preserves the contour of the image.

Example 2: Encrypt a pattern P=(P_1, P_2, P_3)=$\left(\begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\square\\\square&\square\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\right)$ into $P' = \left(P'_1, P'_2, P'_3\right)$ using the proposed 2-XDHS.

Because P is a changeable pattern, A17, according to **Table 2**, we can change this pattern into B5 or B8, with Δ=1. Consider the case of changing it into B5. There is only one pattern $\left(\begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\square&\square\\\square&\square\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\right)$ with the smallest d=1, with $\sum_{j=1}^{3}\left|d/3 - d_j\right| = 4/3$ being the minimum. Consider another case of changing it into B8. There is also one pattern $\left(\begin{smallmatrix}\square&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\square\\\square&\square\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\right)$ with the smallest d=1, with $\sum_{j=1}^{3}\left|d/3 - d_j\right| = 4/3$ being the minimum. We can choose $\left(\begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\square&\square\\\square&\square\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\right)$ or $\left(\begin{smallmatrix}\square&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\square\\\square&\square\end{smallmatrix}\ \begin{smallmatrix}\blacksquare&\blacksquare\\\blacksquare&\blacksquare\end{smallmatrix}\right)$ to make the number of modified pixels in I'_1, I'_2, and I'_3 as similar as possible, i.e., r_1=r_2=r_3=1/3.

3.2 Proposed *k*-XDHS with *k*>2

We have (k+1) halftone images (I_1, I_2, …, I_{k+1}) in a k-XDHS. When considering (H_1, H_2, …, H_{k+1}) in the pattern ($P_{i,1}$, $P_{i,2}$, …, $P_{i,k+1}$), there will be too many changeable and unchangeable patterns in a k-XDHS. Here, we show an approach to construct a k-XDHS from the 2-XDHS. We first partition the k+1 images (I_1, I_2, …, I_{k+1}) into three sets {S_1, S_1, S_3} by **PA**(I_1, I_2, …, I_{k+1}), where $|S_i|\geq 1$ and $|S_1|+|S_2|+|S_3|=(k+1)$. We perform the XOR operation for the images in each set to obtain three noise-like images, and then use them as the inputs of 2-XDHS. Finally, we averagely modify the pixels in the final k+1 modified halftone images (I'_1, I'_2, …, I'_{k+1}) according to the three outputs (two-stego images and one reconstructed image) of the 2-XDHS. The formal encryption **algorithm 3** and decryption **algorithm 4** are described as follows.

Algorithm 3: Encryption of the proposed *k*-XDHS
Input: k halftone cover images I_1–I_k; one halftone secret image I_{k+1}.
Output: $E_{k,k}(I_1, I_2, \ldots, I_{k+1})= I'_1, I'_2, \ldots, I'_k$. /* k stego-images */
1) Obtain a 3-partition {S_1, S_1, S_3} by **PA**(I_1, I_2, …, I_{k+1});
2) Obtain three noise-like halftone images- $O_1= I_1\oplus\ldots\oplus I_{|S_1|}$; $O_2= I_{|S_1|+1}\oplus\ldots\oplus I_{|S_2|+|S_1|}$; $O_3= I_{|S_1|+|S_2|+1},\oplus\ldots\oplus I_{k+1}$;
3) Obtain O'_1, O'_2 and O'_3;
3-1) Get O'_1 and O'_2 by $E_{2,2}(O_1, O_2, O_3)=(O'_1, O'_2)$;
/* note: the probabilities r_1, r_2, and r_3 used in Algorithm 2 are determined as $r_1=|S_1|/(k+1)$, $r_2=|S_2|/(k+1)$, and $r_3=|S_3|/(k+1)$; the chosen probabilities make the modifications averagely distributed in the final images (I'_1, I'_2, …, I'_{k+1} */)
3-2) $O'_3 = O'_1\oplus O'_2$;
4) Obtain the ith blocks, ($P_{i,1}$, $P_{i,2}$, $P_{i,3}$) and $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$, from ($O_1$, O_2, O_3) and (O'_1, O'_2, O'_3), respectively;
5) For j=1 to 3 do {
5-1) For i=1 to (x×y) /4 do {
If $P_{i,j}\neq P'_{i,j}$ then averagely modify the pixels of the images in the set S_j to satisfy $P_{i,j} = P'_{i,j}$; in the meantime, the modifications should satisfy conditions (X-2) and (X-3);}
5-2) Put the modified pixels back into I'_1, I'_2, …, and I'_k };
6) Output k stego-images $E_{k,k}(I_1, I_2, \ldots, I_{k+1})=(I'_1, I'_2, \ldots, I'_k)$.

It is computationally infeasible to directly deal with the changeable and unchangeable patterns of (H_1, H_2, …, H_{k+1}) in a k-XDHS for a large k. Our k-XDHS still uses the changeable and unchangeable patterns in a 2-XDHS. Such construction based on 2-XDHS reduces the complexity order to 2 for any k-XDHS.

Algorithm 4: Decryption of the proposed *k*-XDHS
Input: k stego-images I'_1–I'_k.
Output: $D_{k,k}(I'_1, I'_2, \ldots, I'_k)$.
1) Print out k shadows on transparencies. Stack and align them on an overhead projector;
/* It is possible to use a GIMP image editing tool instead of an overhead projector to superimpose the shadows in decoding */

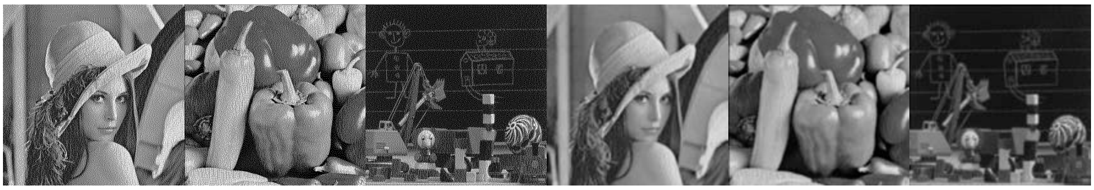
2) Decrypt the secret directly by HVS.

4. Experiment and Security Analysis

4.1 Experimental Results

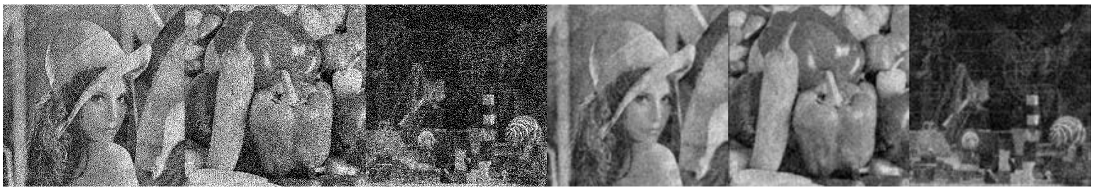
To reasonably evaluate halftone images, we applied a low-pass filter (LPF) (a *Gaussian* LPF with an 11×11 square matrix and a standard deviation of 2.0) to simulate HVS to measure the visual quality. The PSNR of this filtered image is the so-called modified peak signal-to-noise ratio (MPSNR). Example 4 shows the halftone images and the filtered halftone images of the proposed *k*-XDHS and the *randomized* *k*-XDHS for 2≤*k*≤5.

Example 3: Construct the proposed *k*-XDHS and the *randomized* *k*-XDHS, 2≤*k*≤5, respectively. We used five 512×512 halftone images: I_1(Lena), I_2 (Pepper), I_3 (Toy), I_4 (Tank), I_5 (Lake), and I_6 (Jet). In a *k*-XDHS, we used I_1, I_2, …, I_{k+1} images. For example, we used three images, I_1 (*Lena*), I_2 (*Pepper*), and I_3 (*Toy*), in 2-XDHS. All six images will be used in 5-XDHS. **Fig. 3** shows the original halftone images and the halftone images filtered through an LPF. The experimental results of our 2-XDHS are shown in **Fig. 4**. The halftone stego-images are shown in **Fig. 4-(a)** and **Fig. 4-(b),** which reveal their filtered images. **Fig. 5** is the result of the *randomized* 2-XDHS. The MPSNRs of the original halftone images *Lena*, *Pepper*, and *Toy* are 27.82 dB, 26.88 dB, and 27.61 dB, respectively. The proposed 2-XDHS (respectively the *randomized* 2-XDHS) has MPSNRs of 22.90 dB (19.94 dB), 21.90 dB (19.37 dB), and 22.83 dB (17.50 dB) for *Lena*, *Pepper*, and *Toy*, respectively. These MPSNR values are consistent with a real situation.



(a-1) **(a-2)** **(a-3)** **(b-1)** 27.82 dB **(b-2)** 26.88 dB **(b-3)** 27.61 dB

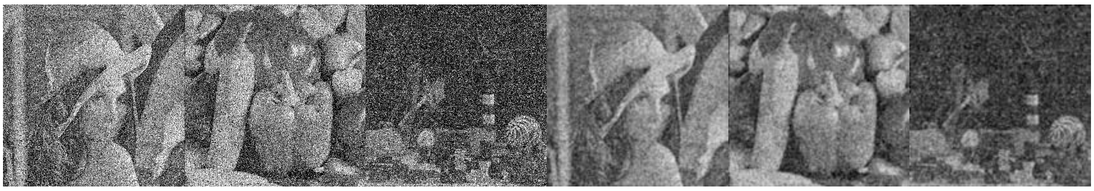
Fig. 3. Three images, I_1 (*Lena*), I_2 (*Pepper*), and I_3 (*Toy*): **(a)** original halftone images and **(b)** filtered halftone images.



(a-1) **(a-2)** **(a-3)** **(b-1)** 22.90 dB **(b-2)** 21.90 dB **(b-3)** 22.83 dB

Fig. 4. Three images, I_1 (*Lena*), I_2 (*Pepper*), and I_3 (*Toy*) using proposed 2-XDHS: **(a)** halftone images and **(b)** filtered halftone images.

The images in **Fig. 4-(a)** really have better visual quality than those in **Fig. 5-(a)**. For example, we still see the curled hair in *Lena* (**Fig. 4-(a-1)**), while the hair is blurred in **Fig. 5-(a-1)**. **Table 3** lists all of the MPSNRs of the original halftone images, the halftone images of the proposed *k*-XDHS, and the halftone images of the randomized *k*-XDHS for 2≤*k*≤5. It can be seen that our schemes have better MPSNR values than the *randomized* schemes. In particular, ours are more effective for k=2. Obviously, the improvement is reduced when k increases because the modifications are averagely distributed among the (k+1) images. The numbers and percentages of modified pixels for the schemes in **Table 3** are shown in **Table 4**. It is observed that even though the proposed XDHS has a greater number of modified pixels than the *randomized* XDHS, our scheme has better MPSNR. For example, there are 62,431 and 43,601 pixels in *Lena* that are modified in the proposed 2-XDHS and the randomized 2-XDHS, respectively. However, **Fig. 4-(b-1)** has a better PSNR (22.90 dB compared to 19.94 dB). This result proves that our block-wise operation effectively minimizes the visual distortion.



(a-1) **(a-2)** **(a-3)** **(b-1)** 19.94 dB **(b-2)** 19.37 dB **(b-3)** 17.50 dB

Fig. 5. Three images, I_1 (*Lena*), I_2 (*Pepper*), and I_3 (*Toy*) using *randomized* 2-XDHS: **(a)** halftone images and **(b)** filtered halftone images.

The proposed k-XDHS uses a block-wise operation to minimize the difference in the *Hamming weight* (condition (**X-2**)) and the difference in the *Hamming distance* (condition (**X-3**)) in a 2×2-pixel grid. Moreover, our k-XDHS satisfies condition (**X-1**) and can decode the secret. In contrast, the *randomized* k-XDHS only satisfies condition (**X-1**) by averagely modifying pixels in (k+1) images. To demonstrate the performance of our k-XDHS, we also compared our k-XDHS with the k-ODHS in [3] and (k, k)-EVCS in [14][18]. In fact, all of the VCSs were simultaneously effective for both OR and XOR decoding operations. Thus, the OR-ed and XOR-ed results of the (k, k)-EVCS are both shown for comparison. Three schemes were used in the experiment: (I) 2-ODHS of Fu et al. [3], (II) (2, 2)-EVCS of Naor et al. [14], and (III) (2, 2)-EVCS of Liu et al. [18]. For these experiments, we used *Lena* and *Pepper* as stego-images and *Toy* as the secret image.

Scheme-I (2-ODHS of Fu et al.):

Fig. 6 shows the two stego-images: *Lena* (27.82 dB) and *Pepper* (26.71 dB). Although, the stego-images have a high MPSNR, we cannot reveal the secret *Toy* image in the stacked result (see **Fig. 6-(a-3)**). Actually, the hidden secret in the 2-ODHS of Fu et al. appears with a "normal" or "lower-than-normal" intensity in the reconstructed images. It is not suitable to hide a natural image. Figs. 6 (b) and (c) show the stacked results when the secret image is a printed letter $\boxed{\text{A}}$ for the same cover image (*Lena*) and different cover images (*Lena* and *Pepper*), respectively. The secret $\boxed{\text{A}}$ is indistinct in **Fig. 6-(c)** because of the effects of the different cover images.

(a-1) (a-2) (a-3) (b) (c)

Fig. 6. 2-ODHS: **(a)** secret image is natural halftone image *Toy* and **(b)** secret image is printed letter $\boxed{A}$ for same cover.

Scheme-II ((2, 2)-EVCS of Naor et al.):

Construct the (2, 2)-EVCS of Naor et al. with *m*=4. The eight base matrices are $B_0^{00}=\begin{bmatrix}1100\\0110\end{bmatrix}$, $B_0^{01}=\begin{bmatrix}1100\\1110\end{bmatrix}$, $B_0^{10}=\begin{bmatrix}1110\\0110\end{bmatrix}$, $B_0^{11}=\begin{bmatrix}1110\\1110\end{bmatrix}$, $B_1^{00}=\begin{bmatrix}1100\\0011\end{bmatrix}$, $B_1^{01}=\begin{bmatrix}1100\\0111\end{bmatrix}$, $B_1^{10}=\begin{bmatrix}1110\\0011\end{bmatrix}$, $B_1^{11}=\begin{bmatrix}1110\\0111\end{bmatrix}$. We used 3B1W (respectively 4B0W) and 2B2W (respectively 3B1W) to represent the black and white pixels in the stego-images (respectively the reconstructed image). Suppose that all of the pixels in the two stego-images and the secret image are black, we should use $B_1^{11}=\begin{bmatrix}1110\\0111\end{bmatrix}$ to expand a secret pixel to 4 sub-pixels. The size of the stego-image is expanded four times.

(a-1) (a-2) (a-3) (b) (c) (d)

Fig. 7. (2, 2)-EVCS of Naor et al.

Fig. 7-(a) shows the OR-ed result, where the MPSNRs of *Lena*, *Pepper*, and *Toy* are 14.32 dB, 13.28 dB, and 13.25 dB, respectively. **Fig. 7-(b)** is the XOR-ed result of **Fig. 7-(a-1)** and **Fig. 7-(a-2)**. **Fig. 7-(c)** and **(d)** are the OR-ed and XOR-ed results when using a printed letter $\boxed{A}$ as the secret. It is observed that (2, 2)-EVCS of Naor et al. is only suitable to hide a simple printed letter image; moreover the XOR-ed result contains the remnant cover images.

Scheme-III ((2, 2)-EVCS of Liu et al.):

Construct (2, 2)-EVCS of Liu et al. with *m*=4. The eight base matrices (2, 2)-EVCS are $B_0^{00}=\begin{bmatrix}1000\\1000\end{bmatrix}$, $B_0^{01}=\begin{bmatrix}1000\\1011\end{bmatrix}$, $B_0^{10}=\begin{bmatrix}1011\\1000\end{bmatrix}$, $B_0^{11}=\begin{bmatrix}1011\\1011\end{bmatrix}$, $B_1^{00}=\begin{bmatrix}1000\\0100\end{bmatrix}$, $B_1^{01}=\begin{bmatrix}1000\\0111\end{bmatrix}$, $B_1^{10}=\begin{bmatrix}1011\\0100\end{bmatrix}$, $B_1^{11}=\begin{bmatrix}1011\\0111\end{bmatrix}$. **Fig. 8-(a)** shows the OR-ed result of (2, 2)-EVCS, where the MPSNRs of the two stego-images, *Lena* and *Pepper*, are 19.46 dB and 17.49 dB, respectively. The OR-ed result (**Fig. 8-(a-3)**) and the XOR-ed result (**Fig. 8-(b)**) are terribly degraded, where the *Toy* image cannot be recognized successfully. **Fig. 8-(c)** and **(d)** are the OR-ed and XOR-ed images when using a printed letter image $\boxed{A}$ as the secret. The (2, 2)-EVCS of Liu et al. produces better visual quality in stego-images than the (2, 2)-EVCS of Naor et al. (*note*: 3B1W and 1B3W for black and white colors in stego-images), but results in a poor reconstructed image.

(a-1) (a-2) (a-3) (b) (c) (d)

Fig. 8. (2, 2)-EVCS of Liu et al.

Table 3. Comparison between proposed XDHS and *randomized* XDHS.

(*k*, *k*)-XISSS \ Halftone (MPSNR)		I_1: Lena (27.82 dB)	I_2: Pepper (26.88 dB)	I_3: Toy (27.61 dB)	I_4: Tank (26.97 dB)	I_5: Lake (24.24 dB)	I_6: Jet (25.58 dB)
k=2	our scheme	22.90 dB	21.90 dB	22.83 dB	–	–	–
	randomized scheme	19.94 dB	19.37 dB	17.50 dB	–	–	–
k=3	our scheme	23.58 dB	23.04 dB	23.91 dB	24.36 dB	–	–
	randomized scheme	21.64 dB	20.76 dB	19.53 dB	22.27 dB	–	–
k=4	our scheme	24.50 dB	23.70 dB	24.51 dB	24.56 dB	21.89 dB	–
	randomized scheme	22.62 dB	21.72 dB	20.87 dB	23.06 dB	20.27 dB	–
k=5	our scheme	25.34 dB	25.00 dB	25.19 dB	25.00 dB	22.91 dB	24.20 dB
	randomized scheme	23.29 dB	22.38 dB	21.80 dB	23.59 dB	20.90 dB	21.50 dB

A comparison of the experimental results for the proposed *k*-XDHS, randomized *k*-XDHS, *k*-ODHS, and (*k*, *k*)-EVCS is summarized in **Table 5**. Our *k*-XDHS produced the best visual quality for stego-images and the reconstructed image. In addition, we could hide the natural halftone image. The other schemes were suitable for hiding the printed-letter image. All of the above schemes provide the feature of viewing the hidden image directly on stego-images.

4.2 Security Analysis

Our *k*-XDHS with *k*>2 is an extension of the proposed 2-XDHS. The *randomized* 2-XDHS uses the bitwise XOR-ed operation and works as a one-time pad. If there is no vulnerability in the randomization process (step (2-1) of **Algorithm 1**, which randomly flips one pixel in *k*+1 pixels when $p_{i,1} \oplus \ldots \oplus p_{i,k} \neq p_{i,k+1}$, it is not possible to gain anything from a stego-image. Therefore, the *randomized* 2-XDHS is unbreakable and clearly secure. Our 2-XDHS is not a one-time pad like the *randomized* 2-XDHS.

Table 4. Number and percentage of modified pixels for proposed *k*-XDHS and *randomized* *k*-XDHS.

(*k*, *k*)-XISSS \ halftone image		I_1: Lena	I_2: Pepper	I_3: Toy	I_4: Tank	I_5: Lake	I_6: Jet
k=2	our scheme	62431 (23.81%)	60300 (23.00%)	64893 (24.75%)	–	–	–
	randomized scheme	43601 (16.63%)	43780 (16.70%)	43671 (16.65%)	–	–	–
k=3	our scheme	44170 (16.84%)	44460 16.9601%	42847 (16.34%)	44539 (16.99%)	–	–
	randomized scheme	32843 (12.52%)	32660 (12.45%)	32757 (12.49%)	32507 (12.40%)	–	–
k=4	our scheme	38175 (14.56%)	37798 (14.41%)	38222 (14.58%)	37815 (14.42%)	38753 (14.78%)	–
	randomized scheme	26333	26355	26067	26407	26051	–

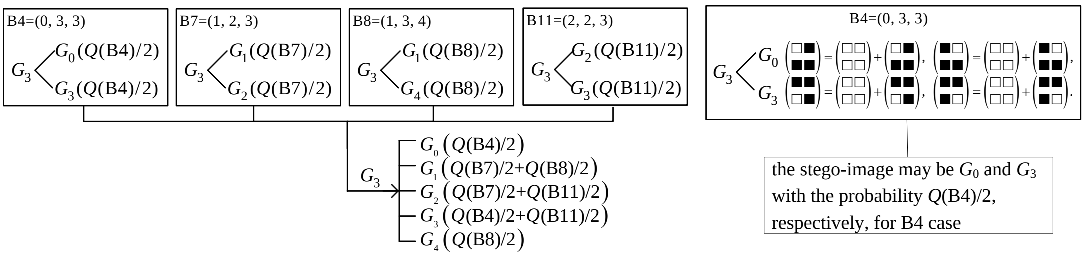
		(10.04%)	(10.05%)	(9.94%)	(10.07%)	(9.93%)	
k=5	our scheme	33851 (12.91%)	32415 (12.36%)	33588 (12.81%)	32649 (12.45%)	33201 (12.66%)	33337 (12.71%)
	randomized scheme	21937 (8.36%)	21878 (8.34%)	21807 (8.31%)	22132 (8.44%)	21599 (8.23%)	21900 (8.35%)

Table 5. Comparison of k-XDHS, k-ODHS, and (k, k)-EVCS.

capability \ scheme	proposed k-XDHS	randomized k-XDHS	k-ODHS	(k, k)-EVCS
visual quality	Excellent	Good	Poor	Poor
secret image	natural image	natural image	printed-text	printed-text
decoding operation	XOR	XOR	OR	OR/XOR
image expansion	NO	NO	NO	YES
easy decoding*	YES	YES	YES	YES

* the hidden image can be viewed directly on stego-images

An attacker could use the prior probabilities to try to compromise the secrecy. We first determine all of the prior probabilities, $Q(j\,|\,i)$, that the grid in a secret image is G_i and the grid in a stego-image is G_j, where $i\in[0, 4]$ and $j\in[0, 4]$. For example, $Q(1\,|\,3)$ denotes the probability that a grid in a secret image has $w(G_3)$=3, while the grid in a stego-image has $w(G_1)$=1.



(a) (b)

Fig. 9. Probability $Q(j|3)$ in proposed 2-XDHS: (a) all $Q(j|3)$, $0\le j\le 4$ (b) B4 case.

The following shows how to determine the probabilities of $Q(j\,|\,3)$, $0\le j\le 4$. By observation, there are only B4, B8, B7, and B11 having the grid G_3. As shown in **Fig. 9**, for the case B4, if a grid in a secret image has $w(G_3)$=3, the grid in the stego-image may have *Hamming weight* $w(G_3)$=3 and $w(G_0)$=0 with the half probability. Consider all four cases (B4, B8, B7, and B11). Then, we have

$$Q(0\,|\,3):Q(1\,|\,3):Q(2\,|\,3):Q(3\,|\,3):Q(4\,|\,3)$$

$$=Q(\mathrm{B4}):(Q(\mathrm{B7})+Q(\mathrm{B8})):(Q(\mathrm{B7})+Q(\mathrm{B11})):(Q(\mathrm{B4})+Q(\mathrm{B11})):Q(\mathrm{B8}),$$

where Q(B4), Q(B7), Q(B8), and Q(B11) are the probabilities of B4, B7, B8, and B11, respectively, using two stego-images and the reconstructed image. The probabilities of $Q(j\,|\,3)$, $0\le j\le 4$ are then calculated as follows.

$$\begin{cases} Q(0\,|\,3) = Q(\mathrm{B4})\big/ Q_3\,; \\ Q(1\,|\,3) = (Q(\mathrm{B7}) + Q(\mathrm{B8}))\big/ Q_3\,; \\ Q(2\,|\,3) = (Q(\mathrm{B7}) + Q(\mathrm{B11}))\big/ Q_3\,; \\ Q(3\,|\,3) = (Q(\mathrm{B4}) + Q(\mathrm{B11}))\big/ Q_3\,; \\ Q(4\,|\,3) = Q(\mathrm{B8})\big/ Q_3\,; \\ \text{where } Q_3 = 2\times(Q(\mathrm{B4}) + Q(\mathrm{B8}) + Q(\mathrm{B7}) + Q(\mathrm{B11})). \end{cases} \tag{1}$$

By the same approach, from **Fig. 10**, the other values of other probabilities, $Q(j\,|\,0)$, $Q(j\,|\,1)$, $Q(j\,|\,2)$, and $Q(j\,|\,4)$, are shown in Eqs. (2)-(5).

$$\begin{cases} Q(0\,|\,0) = Q(\mathrm{B1})\big/ Q_0\,; \\ Q(1\,|\,0) = Q(\mathrm{B2})\big/ Q_0\,; \\ Q(2\,|\,0) = Q(\mathrm{B3})\big/ Q_0\,; \\ Q(3\,|\,0) = Q(\mathrm{B4})\big/ Q_0\,; \\ Q(4\,|\,0) = Q(\mathrm{B5})\big/ Q_0\,; \\ \text{where } Q_0 = Q(\mathrm{B1}) + Q(\mathrm{B2}) + Q(\mathrm{B3}) + Q(\mathrm{B4}) + Q(\mathrm{B5}). \end{cases} \tag{2}$$

$$\begin{cases} Q(0\,|\,1) = 0.5\times Q(\mathrm{B2})\big/ Q_1\,; \\ Q(1\,|\,1) = 0.5\times\big(Q(\mathrm{B2}) + Q(\mathrm{B6})\big)\big/ Q_1\,; \\ Q(2\,|\,1) = 0.5\times\big(Q(\mathrm{B6}) + Q(\mathrm{B7})\big)\big/ Q_1\,; \\ Q(3\,|\,1) = 0.5\times\big(Q(\mathrm{B7}) + Q(\mathrm{B8})\big)\big/ Q_1\,; \\ Q(4\,|\,1) = 0.5\times Q(\mathrm{B8})\big/\big(Q(\mathrm{B2}) + Q(\mathrm{B6}) + Q(\mathrm{B7}) + Q(\mathrm{B8})\big); \\ \text{where } Q_1 = \mathrm{Q(B2)+Q(B6)+Q(B7)+Q(B8)}. \end{cases} \tag{3}$$

$$\begin{cases} Q(0\,|\,2) = 0.5\times Q(\mathrm{B3})\big/ Q_2\,; \\ Q(1\,|\,2) = \big(Q(\mathrm{B6}) + 0.5\times Q(\mathrm{B7})\big)\big/ Q_2\,; \\ Q(2\,|\,2) = \big(0.5\times(Q(\mathrm{B3}) + Q(\mathrm{B9})) + Q(\mathrm{B10})\big)\big/ Q_2\,; \\ Q(3\,|\,2) = \big(0.5\times Q(\mathrm{B7}) + Q(\mathrm{B11})\big)\big/ Q_2\,; \\ Q(4\,|\,2) = 0.5\times Q(\mathrm{B9})\big/ Q_2\,; \\ \text{where } Q_2 = \mathrm{Q(B3)+Q(B6)+Q(B7)+Q(B9)} + Q(\mathrm{B10}) + Q(\mathrm{B11}). \end{cases} \tag{4}$$

$$\begin{cases} Q(0\,|\,4) = 0.5\times Q(\mathrm{B5})\big/ Q_4\,; \\ Q(1\,|\,4) = 0.5\times Q(\mathrm{B8})\big/ Q_4\,; \\ Q(2\,|\,4) = Q(\mathrm{B9})\big/ Q_4\,; \\ Q(3\,|\,4) = 0.5\times Q(\mathrm{B8})\big/ Q_4\,; \\ Q(4\,|\,4) = 0.5\times Q(\mathrm{B5})\big/ Q_4\,; \\ \text{where } Q_4 = Q(\mathrm{B5}) + Q(\mathrm{B8}) + Q(\mathrm{B9}). \end{cases} \tag{5}$$

There are a total of 35 patterns, and thus the probability of occurrence for each pattern (A-patterns and B-patterns) is 1/35. A-patterns will be modified into B-patterns (see step (2-1) of Algorithm 2). Thus, we only have B-patterns.

From **Table 2**, B1 may come from A8, A9, and A10 with the probability 1/3×1/35 and from A16 with the probability 1/4×1/35. Finally, Q(B4)=1/35+1/3×1/35+1/3×1/35+1/4×1/35=9/140. By the same approach, we have Q(B7)=23/140, Q(B8) $=14\frac{1}{3}/140$, and Q(B11) $=19\frac{1}{3}/140$. From Eq. (1), and the values of

Q(B4), Q(B7), Q(B8), and Q(B11), we determine $Q(0 \mid 3)$ = 6.8%. All of the prior probabilities, $Q(j \mid i)$, are shown in **Table 6**.

We now show how an attacker could use the prior probabilities for cryptanalysis. We first precisely define the scope of the secrecy ensured by our proposed XDHS. An attacker's knowledge is described as follows. He has the detailed procedure of **Algorithm 2**, but does not have the two pieces of randomization information of step (2-1) and step (2-3) in **Algorithm 2**. The first piece of randomization information is that an attacker does not actually know whether the B-pattern comes from an A-pattern or was originally a B-pattern.

Table 6. Probabilities of $Q(j \mid i)$ for $0 \le i, j \le 4$.

$Q(j \mid i)$	i=0	i=1	i=2	i=3	i=4
j=0	14.5%	8.2%	4.2%	6.8%	9.8%
j=1	27.4%	23.0%	34.6%	28.4%	19.2%
j=2	18.5%	31.4%	19.2%	32.3%	42.0%
j=3	21.8%	27.0%	33.5%	21.6%	19.2%
j=4	17.8%	10.4%	8.5%	10.9%	9.8%

The second piece of randomization information is that an attacker has no information about which pixel is modified when the pattern is modified to satisfy condition (**X-1**). The argument that an attacker cannot gain a secret image by using the prior probabilities is reasonable because of the following rationales.

(a) The *Hamming weight* in grid G_j of a secret image may be changed in step (2-1) of **Algorithm 2** (note: a B-pattern may come from an A-pattern).

(b) Even though an attacker obtains a correct *Hamming weight* for grid G_j, he cannot get the correct arrangement.

(c) It seems that an attacker can obtain a secret image by using the following approach. An attacker regards G_0 and G_1 as white areas, and G_3 and G_4 as black areas; also half of G_2 is regarded as a black area and half as a white area. From **Table 6**, all of the $Q(0 \mid i) + Q(1 \mid i)$ and $Q(3 \mid i) + Q(4 \mid i)$, $0 \le i \le 4$ are calculated.

$$\begin{cases} Q(0 \mid 0) + Q(1 \mid 0) = 41.9\%, \ Q(3 \mid 0) + Q(4 \mid 0) = 39.6\%; \\ Q(0 \mid 1) + Q(1 \mid 1) = 31.2\%, \ Q(3 \mid 1) + Q(4 \mid 1) = 37.4\%; \\ Q(0 \mid 2) + Q(1 \mid 2) = 38.8\%, \ Q(3 \mid 2) + Q(4 \mid 2) = 42.0\%; \\ Q(0 \mid 3) + Q(1 \mid 3) = 35.2\%, \ Q(3 \mid 3) + Q(4 \mid 3) = 32.5\%; \\ Q(0 \mid 4) + Q(1 \mid 4) = 29.0\%, \ Q(3 \mid 4) + Q(4 \mid 4) = 29.0\%; \end{cases} \tag{6}$$

By (6), $Q(0 \mid i) + Q(1 \mid i)$ and $Q(3 \mid i) + Q(4 \mid i)$ are almost the same. Thus, it is not possible to obtain the secret image from a stego-image using the above approach. Suppose an attacker adopts this approach to recover the secret images in a 2-XDHS. The cover images are 512×512 halftone images, Lena (I_1) and Pepper (I_2), and the secret image is Toy (I_3). A stego-image (Lena) is given. As shown in **Fig. 11**, the reconstructed image for I_3 from this attack is noise-like. Thus, it is impossible to visually decode the secret.

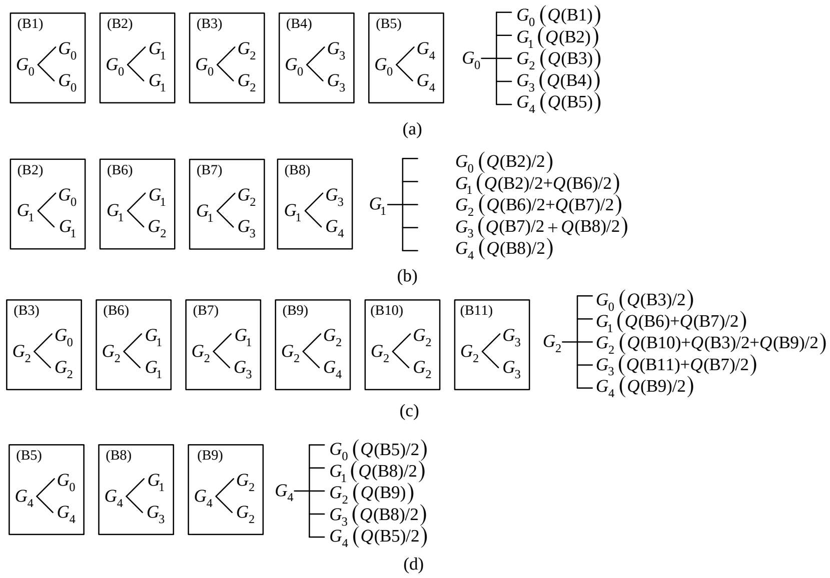


Fig. 10. Probabilities $Q(j|0)$, $Q(j|1)$, $Q(j|2)$, and $Q(j|4)$ in proposed 2-XDHS: (a) $Q(j|0)$, (b) $Q(j|1)$, (c) $Q(j|2)$, and (d) $Q(j|4)$.

In a 2-XDHS, there are only three images (two stego-images (I_1, I_2) and one secret image (I_3)). There is a high probability of $P_{i,1} \oplus P_{i,2} = P_{i,3}$. Suppose that a secret is an entirely white background. This probability will even be higher, and allows one to visually reveal the boundary artifacts around a stego-image in the other stego-image.

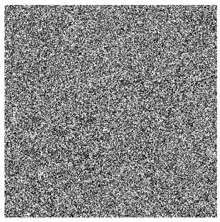
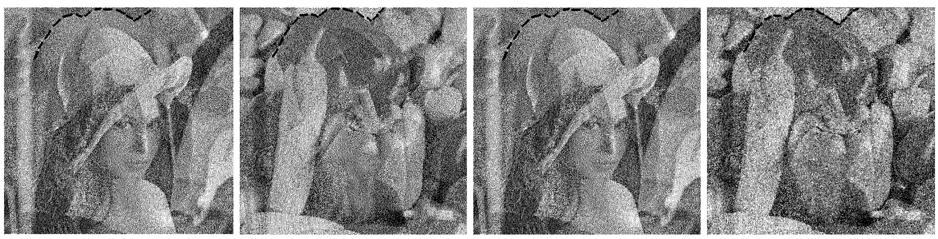


Fig. 11. Reconstructed image by regarding G_0 and G_1 (respectively G_3 and G_4) as white (respectively black), and randomly choosing black or white for G_2.

Fig. 12-(a) shows the visible boundary around the I_2 in I_1, where I_1 and I_2 are two stego-images, *Lena* and *Pepper*, when using an entirely white image as a secret image. The appearance of boundary artifacts also occurs in the *randomized* 2-XDHS (see **Fig. 12-(b)**). However, this information leakage will diminish as k increases.



(a-1) **(a-2)** **(b-1)** **(b-2)**

Fig. 12. Visible boundary artifacts around I_2 in I_1, where I_1 and I_2 are two stego-images, *Lena* and

Pepper, when using an entirely white image as a secret image: **(a)** proposed 2-XDHS and **(b)** *randomized* 2-XDHS.

Fig. 13 and **14** show that the proposed (4, 4)-XISSS and *randomized* (4, 4)-XISSS do not have visible boundary artifacts even when using an entirely white secret image. In relation to data hiding, imperceptibility is a major essential consideration. Step (2) in **Algorithm 2** can be modified as step (2′) to completely solve this problem of information leakage:

2′) For i=1 to $(x \times y)$ /4 do {

2′-1) If $(P_{i,1}, P_{i,2}, P_{i,3}) \in$[B1-B11] then go to step (2-2)

else $(P_{i,1}, P_{i,2}, P_{i,3})= \mathbf{M}\left(P_{i,1}, P_{i,2}, P_{i,3}\right)$;

2′-2) Obtain all $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$ by $\mathbf{P}\left(P_{i,1}, P_{i,2}, P_{i,3}\right)$;

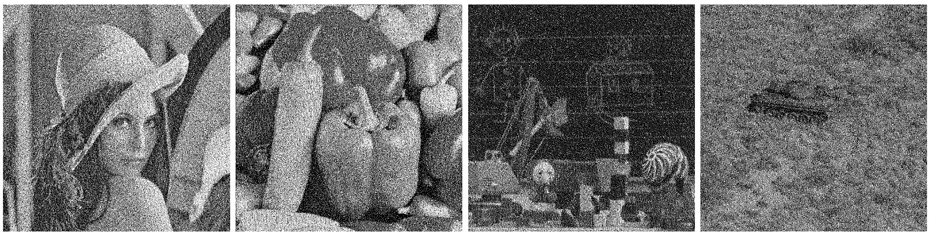
2′-3) Do Case 1 with probability r_0, and do Case 2 with probability $(1- r_0)$;

Case 1: Randomly choose one pattern $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$ in all patterns $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$;

Case 2: Find a pattern $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$ having the smallest $d = d_1 + d_2 + d_3$ with $\sum_{j=1}^{3}\left|d/3 - d_j\right|$ being minimum (note: if more than one pattern, choose a $\left(P'_{i,1}, P'_{i,2}, P'_{i,3}\right)$ with r_1=r_2= r_3=1/3);

/* even though an original pattern $(P_{i,1}, P_{i,2}, P_{i,3})$ satisfies (X-1) and has Δ=d=0, it will be modified with probability r_0. */

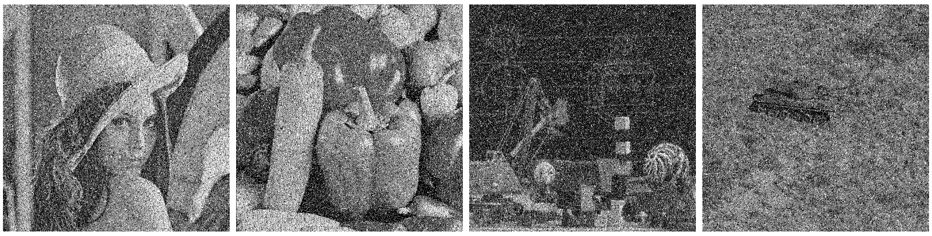
2′-4) Put $\left(P'_{i,1}, P'_{i,2}\right)$ back to I'_1 and I'_2; };



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

Fig. 13. Four stego-images of proposed 4-XDHS, *Lena*, *Pepper*, *Toy*, and *Tank*, when using entirely white image as secret image.

The visible boundary artifacts can be avoided by adjusting threshold probability r_0, which can control the trade-off between the visual quality of all of the images and the remnant boundary artifacts. Notice that step (2′) is reduced to step (2) for r_0=0. Actually, this information leakage will not occur when exclusively using natural images for all of the images, rather than the extreme case of using an entirely white (or black) image.



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

Fig. 14. Four stego-images of randomized 4-XDHS, *Lena*, *Pepper*, *Toy*, and *Tank*, when using an

entirely white image as a secret image.

5. Conclusions

This paper proposed a novel k-XDHS to hide a natural halftone image in k natural halftone images. The XDHS was shown to minimize the difference in visual quality between the stego-image and the cover image (a natural image). When k-XDHS was compared with the *randomized* XDHS, our block-wise approach satisfied condition (**X-2**) (minimize the *Hamming weight*) and condition (**X-3**) (minimize the *Hamming difference*). Our scheme had better MPSNR than the *randomized* scheme, and obtained more effective performance for k=2. Moreover, our k-XDHS produced the best visual quality in stego images, compared to k-ODHS and (k, k)-EVCS. In addition, it can hide a natural halftone image. In order to increase the security of our scheme, we used 35 patterns. Thus, the probability of occurrence for each pattern (A-patterns and B-patterns) was 1/35. A thorough security analysis showed that even if an attacker could gain the prior probabilities for an attack, he could not reveal a secret image by using these probabilities. Finally, our XDHS effectively minimized the visual distortion, demonstrated high MPSNRs for stego images, and achieved a strong level of secrecy.

Acknowledgement

This work was supported in part by the Testbed@TWISC, National Science Council under the Grants NSC 99-2219-E-006-011.

References

[1] C.C. Lin and W. H. Tsai, "Secret multimedia information sharing with data hiding capacity by simple logic operations," in *Proc. of 5th World Multi-conference on Systemics, Cybernetics, and Informatics, Vol. I: Information Systems Development*, pp. 50-55, 2001. Article (CrossRef Link)

[2] M.S. Fu and O.C. Au, "Data hiding in halftone image by stochastic error diffusion," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1965-1968, 2001. Article (CrossRef Link)

[3] M.S. Fu and O.C. Au, "Steganography in halftone images: conjugate error diffusion," *Signal Processing*, vol. 83, pp. 2171-2178, 2003. Article (CrossRef Link)

[4] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," *IEEE Trans. on Multimedia*, vol. 6, pp. 528-538, 2004. Article (CrossRef Link)

[5] P. Sherry and A. Savakis, "Improved techniques for watermarking halftone images," *Acoustics, Speech, and Signal Processing*, vol. 5, pp. 1005-1008, 2004. Article (CrossRef Link)

[6] M. Wu, J. Fridrich, M. Goljan and H. Gou, "Handling uneven embedding capacity in binary images: a revisit," *SPIE Conference on Security, Watermarking and Steganography*, vol. 5681, pp. 194-205, 2005. Article (CrossRef Link)

[7] H. Gou and M. Wu, "Improving embedding payload in binary images with super-pixels," in *Proc. of IEEE International Conference on Image Processing*, vol.3, pp.III-277-III-280, 2007. Article (CrossRef Link)

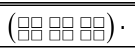
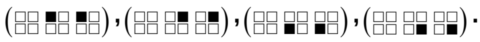
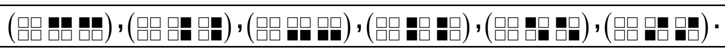
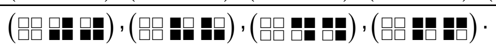
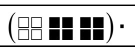
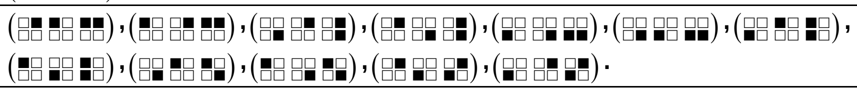
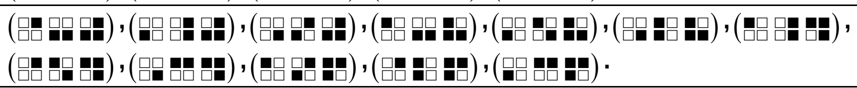
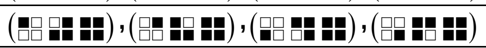
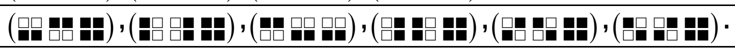
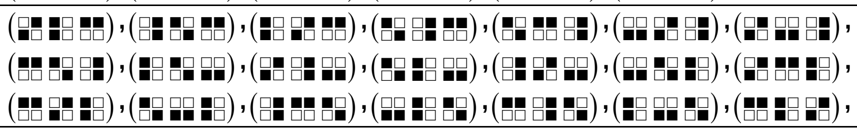
[8] I.S. Lee and W.H. Tsai, "A dynamic-programming approach to data hiding in binary image using block pattern coding with distortion minimization," *IEICE Transaction on Information and Systems*, vol. E90-D, no. 8, pp. 1142-1150, 2007. Article (CrossRef Link)

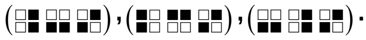
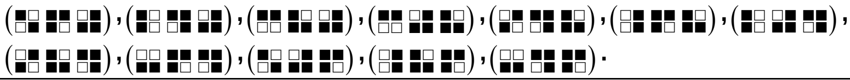
[9] Y.A. Ho, Y.K. Chan, H.C. Wu and Y.P. Chu, "High-capacity reversible data hiding in binary images using pattern substitution," *Computer Standards & Interfaces*, vol. 31, pp. 787-794, 2009. Article (CrossRef Link)

[10] C.L. Tsai, H.F. Chiang, K.C. Fan and C.D. Chung, "Reversible data hiding and lossless reconstruction of binary images using pair-wise logical computation mechanism," *Pattern Recognition*, vol. 38, pp. 1993-2006, 2005. Article (CrossRef Link)
[11] M. Nakajima and Y. Yamaguchi, "Enhancing registration tolerance of extended visual cryptography for natural images," *Journal of Electronic Imaging*, vol. 13, pp. 654–662, 2004. Article (CrossRef Link)
[12] D.Q. Viet and K. Kurosawa, "Almost ideal contrast visual cryptography with reversing," *Cryptology- CT-RSA*, pp. 353-365, 2004. Article (CrossRef Link)
[13] S. Cimato, A. De Santis, A.L. Ferrara and B. Masucci, "Ideal contrast visual cryptography schemes with reversing," *Information Processing Letters*, vol. 93, pp. 199-206, 2005. Article (CrossRef Link)
[14] C.N. Yang, C.C. Wang and T.S. Chen, "Visual cryptography schemes with reversing," *The Computer Journal*, vol. 51, pp. 710-722, 2008. Article (CrossRef Link)
[15] M. Naor and A. Shamir, "Visual Cryptography," *Advances in Cryptology- EUROCRYPT'94*, pp. 1-12, 1994. Article (CrossRef Link)
[16] G. Ateniese, C. Blundo, A. De Santis and D.R. Stinson, "Extended capabilities for visual cryptography," *Theoretical Computer Science*, vol. 250, pp. 143-161, 2001. Article (CrossRef Link)
[17] C.N. Yang and T.S. Chen, "Extended visual secret sharing schemes: improving the shadow image quality," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21, pp. 879-898, 2007. Article (CrossRef Link)
[18] P. Tuyls, H.D.L. Hollmann, J.H. Van Lint and L. Tolhuizen, "XOR-based visual cryptography schemes," *Designs, Codes and Cryptography*, vol. 37, no. 1, pp. 169-186, 2005. Article (CrossRef Link)
[19] F. Liu, C.K. Wu and X.J. Lin, "Some extensions on threshold visual cryptography schemes," *The Computer Journal*, vol. 53, pp. 107-119, 2010. Article (CrossRef Link)

Appendix

Table A-1. Permutation of patterns B1-B11 holding $P_{i,1} \oplus P_{i,2} = P_{i,3}$.

pattern (H_1, H_2, H_3)	permute 4 binary pixels in a 2×2-pixel grid **P** ($P_{i,1}$, $P_{i,2}$, $P_{i,3}$)
B1	
B2	
B3	
B4	
B5	
B6	
B7	
B8	
B9	
B10	

	
B11	

Ching-Nung Yang was born on May 9, 1961 in Kaohsiung, Taiwan. He received the B.S. degree in 1983 and the M.S. degree in 1985, both from the Department of Telecommunication Engineering at National Chiao Tung University. He received the Ph.D. degree in Electrical Engineering from National Cheng Kung University in 1997. During 1987-1989 and 1990-1999, he worked at the Telecommunication Lab. and Training Institute Kaohsiung Center, Chunghwa Telecom Co., Ltd., respectively. He is presently a professor in the Department of Computer Science and Information Engineering at National Dong Hwa University. He is also an IEEE Senior Member. His research interests include coding theory, information security, and cryptography.

Guo-Cin Ye received his M.S. degree in 2009 from the Department of Computer Science and Information Engineering at National Dong Hwa University, Taiwan. He currently serves in the alternative military service.

Cheonshik Kim received his B.S. degree in Computer Engineering from Anyang University, Korea, in 1995; his M.S. degree in Information Engineering from Hankuk University of Foreign Studies (HUFS), Korea, in 1997; and his Ph D. degree in Computer Engineering from HUFS in 2003. He joined the faculty of Sejong University, Korea, where he is currently a professor in the Dept. of Computer Engineering. His research interests include Multimedia systems and Data Hiding. He is a member of IEEK and IEEE.