Hiding Data in Binary Images

Chin-Chen Chang[1], Chun-Sen Tseng[1], and Chia-Chen Lin[2]

[1] Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi 621,
Taiwan, R.O.C.
{ccc, tcs92}@cs.ccu.edu.tw
[2] Department of Computer Science and Information Management
Providence University,
Taichung 433, Taiwan, R.O.C.
mhlin3@pu.edu.tw

Abstract. This paper presents a novel scheme for embedding secret data into a binary image. In Tseng et al.'s scheme, a random binary matrix and a weight matrix are used as the secret keys to protect the secret information. In our scheme, we use a serial number matrix instead of a random binary matrix to reduce computation cost and to provide higher security protection on hidden secret data than Tseng et al. do. Given a cover image divided into blocks of m×n pixels each, our new scheme can hide $\lfloor \log_2(mn+1) \rfloor$ bits of hidden data with one modified bit at most in each block in the cover image. In addition, the hiding capacity of our new scheme offers is as large as that of Tseng et al.'s scheme, but we support higher stego-image quality than Tseng et al.'s scheme does.

1 Introduction

The rapid advancement of the Internet technology has made the Internet the most popular channel for digital data exchanges. Generally speaking, digital data transmitted on the Internet can be in the form of text messages, images, or audio and video files. Despite the convenience the Internet offers for data exchange, one major problem occurs. That is, the data on the Internet is easily tampered with and stolen by attackers during the transmission. In order to deal with this problem, two strategies have been proposed: cryptography and steganography [1-16]. The former strategy usually transfers data into a set of seemingly meaningless codes. Only the authorized user can transform it back to its original form by using some secret information. Many famous encryption schemes, such as RSA, DES, and so on, are already widely accepted commercially. However, the meaningless appearance may be a clue to an unauthorized user and shows that there might be something interesting hidden inside.
The other strategy, called steganography or data hiding, hides a secret message in a cover media to avoid arousing attackers' attention. For example, the outline of a computer motherboard can be embedded into Vincent van Gogh's famous painting "The Starry Night" to fool attackers so that it can be transmitted on the Internet unnoticed. The concept of steganography is similar to the concept of camouflage, which is

Robert H. Deng et al. (Eds.): ISPEC 2005, LNCS 3439, pp. 338 – 349, 2005.

used by animals to protect them from being attacked. Generally, the objective of steganography is to hide a secret message well enough so that unauthorized users will not even be aware of its existence. Several steganographic schemes have been developed to solve the privacy problem [1-16]. In general, the steganography approach can be classified into three categories. In the first category, the schemes hide a secret message in the spatial domain of the cover image [3, 10, 13, 14, 16]. In Lee and Chen's scheme [16], the least significant bit (LSB) of each pixel in the cover image is modified to embed the secret message. In Wang et al.'s scheme [14], the optimal substitution of LSB is exploited. In addition, Chung et al. offered a singular value decomposition (SVD)-based hiding scheme [10], and Tsai et al. exploited the bit plane of each block truncation coding (BTC) block to embed the secret message [13]. In the second category, the schemes embed a secret message into a transformed cover image [1, 2, 4, 8]. Several transformation functions, such as the discrete cosine transformation (DCT) and the discrete wavelet transformation (DWT), are widely used. For example, in Chang et al.'s scheme [1], the middle frequency coefficients of the DCT transformed cover image are employed to embed the secret message. In addition, the quantization table of JPEG is modified to protect the embedded secret message. In Kobayashi et al.'s scheme [2], a secret message is hidden in the JPEG encoded bit streams. In the third category, several schemes that work on index-based cover images [5, 7, 11]. In fact, index-based images such as vector quantization (VQ)-based images, color quantization (CQ)-based (palette-based) images, are commonly used.

However, most cover images of the above schemes are gray-level images or color images. The binary image is not often used to be a cover media [6, 9, 12, 15, 17, 18]. The major reason is that the modification is easily detected when a single pixel is modified in a binary image. In [6], Chen et al. decomposed an image into many blocks first. Then, they divided each block into several non-overlapping subgroups, called partitions. Finally, they come up with the characteristic value of each partition to decide where to hide the secret data. However, they can only conceal just one bit in a 4×4 block. To improve the hiding capacity, Tseng et al.'s scheme divides a cover image into many non-overlapping blocks and hides data in each block [15]. They also generated two matrices, a binary matrix and a weight matrix, to decide which bits need to be modified so that the secret information can be hidden and the good image quality of stego-image can be achieved. Given an m×n block from the cover image, Tseng et al.'s scheme can conceal bits of data in a block by changing two pixels at most.

Since changing any pixel in a binary image can cause a detectable change in the cover binary image, it is important to reduce the number of modified pixels. Otherwise, the steganography is easily detected by the human visual system. With our new scheme, we can hide as many bits as Tseng et al.'s scheme can, and there is only one pixel at most that is modified in each block. The image quality of the cover image is thus improved, and the hidden information is well protected.

The remaining text of this paper is organized as follows. In Section 2, we shall introduce our proposed serial matrix first and then present our proposed scheme. In Section 3, we shall analyze the security of the proposed scheme. Section 4 will discuss our experimental results and compare our performance with Tseng et al.'s. Finally, Section 5 presents the conclusions.

2 Proposed Data Hiding Scheme for Binary Images

To maintain good image quality of stego-image and to reduce the number of modified pixels when hiding secret message in a block, we propose a serial number matrix here to decide which pixel needs to be modified. The proposed serial number matrix is our key technique. Therefore, we will introduce the proposed serial number matrix in Subsection 2.1 and then proves its function in Subsection 2.2. At last, we will present our proposed scheme in Subsection 2.3

2.1 The Proposed Serial Number Matrix

In our proposed scheme, we try to embed $\lfloor \log_2(mn+1) \rfloor$ bits into an $m \times n$ block in a cover image by changing one pixel at most in each block. In other words, using our proposed data hiding scheme, the better case is that none of the pixels needs to be modified, and the worst case of the two is that one pixel in a block needs to be changed. Whichever the case, we always need a good selection mechanism to pick out the pixels to be modified so that the image distortion can be reduced while keeping high stego-image quality and high hiding capacity. This is quite a challenging task. In this paper, we offer a serial number matrix O as our selection mechanism, which can help us to change only one pixel at most in a block. In our proposed serial number matrix O sized $m \times n$, at most $(m \times n-1)$ non-duplicate integers appear. Assume b is a number that shows up in the serial number matrix O, and the values in different positions of the serial number matrix O can be duplicated. Let H_j be the value of the jth secret bit, j=1, 2,…, $\lfloor \log_2(mn+1) \rfloor$. According to the proposed serial number matrix O, the general hiding equation is as follows.

$$H_j = \sum_{i \in N_j} p_i(x) \text{ ,} \tag{1}$$

$O =$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	1

(a)

$O_b =$

0001	0010	0011	0100
0101	0110	0111	1000
1001	1010	1011	1100
1101	1110	1111	0001

(b)

$A =$

1	1	0	1
0	1	0	1
1	0	1	0
0	1	0	0

(c)

$A' =$

1	0	0	1
0	1	0	1
1	0	1	1
0	1	0	0

(d)

Fig. 1. (a) An example of the proposed serial number matrix O sized 4×4, (b) the binary representation of O named O_b, (c) an example binary cover image A sized *4×4*, and (d) a stego-image A' that is the embedding result of the cover image A

where $p_i(x)$ is the pixel value of binary cover image A, which corresponds to serial number x in position i in the serial number matrix O when the jth bit value of the binary representation of x is 1, where $_{i \in N_j}$, $1 \le x \le \lfloor \log 2(mn+1) \rfloor$. N_j is the serial number set which is defined by proposed rule with corresponding H_j, we describe the rule for grouping serial number into each N_j in the following example.

In order to visualize how the above matrix works, let us take a 4×4 image block for example. The proposed serial number matrix O sized 4×4 contains 15 non-duplicate integers shown in Fig. 1(a). The numbers that show up in the serial number matrix are from 1 to 15, and the value in the position 1 in the serial number matrix O is 1, which is the same as the value in position 16.

Assume that we want to hide 4 bits in the binary cover image A shown in Fig. 1(c). According to the proposed serial number matrix O shown in Fig. 1(a) and the general hiding equation shown in Equation (1), four hiding equations are constructed as follows.

$$H_1=p_1(1)+p_3(3)+p_5(5)+p_7(7)+p_9(9)+p_{11}(11)+p_{13}(13)+p_{15}(15)+p_{16}(1). \quad (2)$$
$$H_2=p_2(2)+p_3(3)+p_6(6)+p_7(7)+p_{10}(10)+p_{11}(11)+p_{14}(14)+p_{15}(15). \quad (3)$$
$$H_3=p_4(4)+p_5(5)+p_6(6)+p_7(7)+p_{12}(12)+p_{13}(13)+p_{14}(14)+p_{15}(15). \quad (4)$$
$$H_4=p_8(8)+p_9(9)+p_{10}(10)+p_{11}(11)+p_{12}(12)+p_{13}(13)+p_{14}(14)+p_{15}(15). \quad (5)$$

Based on the general hiding equation shown in Equation (1), each value of H_j (for j = 1, 2, 3, 4) is either odd or even. Besides, we can make sure there will be any two of hiding equations share at least one $p_i(x)$, any three of hiding equations share at least one $p_i(x)$, and any four of hiding equations share at least one $p_i(x)$. Still, each of these four equations also has at least one $p_i(x)$ which will not appear in any other equations. For example, in Equation (2), the serial numbers in positions 1 and 16 in the serial number matrix O are all 1, where the *1*st bit value of their binary representations are 1, and the remaining bit values of their binary representations are 0. Therefore, the corresponding values $p_1(1)$ and $p_{16}(1)$ only appear in Equation (2). The serial number of position 3 in serial number matrix O is 3, where both the first and second bit values of its binary representation are 1. Therefore, its $p_3(3)$ appear in Equations (2) and (3). To sum up, we can modify only one component in each equation to adjust each H_j from odd to even or from even to odd using the above four equations. Based on the above arrangement, first, we can get the corresponding h_j for each H_j by using Equation (6). Then, we can compare h_j with the jth bit value of the secret data s_j to generate R_j using the principle listed in Equation (7). Next, we put R_j, j=4, 3, 2 and 1 together to generate a stream R and then convert the stream R into its decimal representation. Finally, we can obtain the modification position in the block.

$$h_j = H_j \bmod 2, \text{ for } j = 1, 2, 3, \text{ and } 4, \quad (6)$$
$$R_j=0, \text{ if } h_j=s_j$$
$$R_j=1, \text{ otherwise.} \quad (7)$$

With our proposed serial number matrix O, the largest hiding capacity of a block sized m×n is the same as the number of hiding equations generated by the serial number matrix O. In other words, when a 4×4 serial number matrix O contains 15 non-duplicate integers, we can construct four hiding equations according to the general hiding equation shown in Equation (1), and that means the largest hiding capacity of each block is 4 bits.

To describe the function of our proposed serial number matrix O, a simple example is presented as follows. Give a block A as shown in Fig. 1(c). Assume that the secret data is "1001". Fig. 1(a) is an example of our serial number matrix O, and its binary representation is shown in Fig. 1(b). Table 1 shows which pixel needs to be modified, and Fig. 1(d) is the embedding result of block A. Let us take the first bit of the secret data for example. First, we can generate equation H_1 by using Equation (1) and then get the result H_1 (H_1=p_1(1)+p_3(3)+p_5(5)+p_7(7)+p_9(9)+p_{11}(11)+p_{13}(13)+p_{15}(15)+p_{16}(1) =1+0+0+0+1+1+0+0+0+0=3*)*. We can obtain h_1 =1 using Equation (6), and R_1 =0 according to Equation (7). The same operations are conducted to obtain R_2, R_3 and R_4, respectively, shown in Table 1. Finally, we put R_j, for j=4, 3, 2 and 1 together to generate an R stream as "1100". After converting the R stream into its decimal representation, we can obtain $12_{(10)}$. It means that we only need to modify position 12 from 0 to 1 in block A to hide the secret data "1001". The embedding result is A' shown in Fig. 1(d).

Table 1. The secret data, modified H_j's and their corresponding h_j's and R_j's, and the modification position given by the serial number matrix O

Block	s_4,s_3,s_2,s_1	H_4,H_3,H_2,H_1	h_4,h_3,h_2,h_1	R_4,R_3,R_2,R_1	Modification Position *(MP)*
A	1, 0, 0, 1	4, 3, 4, 3	0, 1, 0, 1	1, 1, 0, 0,	$12=2^3*1+2^2*1+2^1*0+2^0*0$

Since the modification position is 12, p_{12}(12) appears in H_3 and H_4 simultaneously. After modification, the value of the modified H_j's are 3, 4, 4, 5, respectively, where j=4, 3, 2 and 1. The new h_j's and R_j's are generated according to Equations (6) and (7). Please note that each new h_j is the same as its corresponding secret data s_j, and no pixel needs to be changed. The modified H_j's and new h_j's and R_j's, where j=4, 3, 2 and 1, are listed in Table 2.

Table 2. The secret data, modified H_j's, their corresponding h_j's and R_j's, and the modification position given by the serial number matrix O

Block	s_4,s_3,s_2,s_1	H_4,H_3,H_2,H_1	h_4,h_3,h_2,h_1	R_4,R_3,R_2,R_1	Modification Position *(MP)*
A	1, 0, 0, 1	**5**, **4**, 4, 3	**1**, **0**, 0, 1	**0**, **0**, 0, 0,	$0=2^3*0+2^2*0+2^1*0+2^0*0$

2.2 Verification of Our Proposed Serial Number Matrix

In the previous subsection, we have illustrated how our serial number matrix O functions. In addition, we claim that the proposed serial number matrix O can guarantee that only one pixel at most is needed in a block to be modified to embed r bits, and r is the number of hiding equations generated by the proposed general hiding equation and serial number matrix O. For example, given a serial number matrix O sized 4×4 with 15 non-duplicate integers, the serial number matrix O can generate four hiding equations at most, which means the largest hiding capacity of each block in the cover image is 4 bits. Before we give more details as to our proposed hiding scheme based on the serial number matrix O, we will try to prove our claim in this subsection.

We claim that if there are 2^r-1 elements in the serial number matrix O that are non-duplicate, then as many as r hiding equations can be generated according to our proposed general hiding equation mentioned in the previous subsection. Besides, each of hiding equations has at least one element of a block which will not appear in any other equations, any two of hiding equations share at least one element of a block, any three of hiding equations share at least one element of a block, and so on. Please refer to Equations (2)-(5).

In this case, do we only need to explore 2^r-1 pixels of a block in a cover image if we want to hide r bits of secret data into a block? Our proof is quite straightforward. We list all the possible modification solutions to check whether the maximum number of modification pixels is 2^r-1.

$$\sum_{c=1}^{r} C_c^r = C_1^r + C_2^r + ... + C_{r-1}^r + C_r^r \cdot \tag{8}$$

Here, c stands for the number of pixels to be modified to hide the secret data. In other words, c is also the number of the modified hiding equations. Since c equals 0, which means that there is no hiding equation, C_0^r is not included in Equation (8). C_1^r means only one h_j, where $1 \le j \le r$, is different from its relative secret bit s_j, $1 \le j \le r$, and only one hiding equation needs to be modified. Since each of hiding equations has at least one element of a block, which will not appear in any other equations, in this case, we only need to modify one pixel of a block in the cover image to achieve our goal. C_2^r means two h_j's, where $1 \le j \le r$, are different from their relative secret bits s_j's, where $1 \le j \le r$, and two hiding equations need to be modified. According to the designing principle of our hiding equations, any two of the hiding equations share one common pixel. Therefore, we can still change one pixel to modify the values of *h's* and then achieve our goal. In all the other cases, the same logic applies. Finally, we come to this conclusion: If we want to hide r bits in a block, we only need to explore $2^r - 1$ ($\sum_{c=1}^{r} C_c^r = 2^r - 1$) pixels at most in our hiding equations and change one pixel at most in any case. Since only $2^r - 1$ pixels need to be explored in each block, the maximum amount of embedded bits is $\lfloor \log_2(mn+1) \rfloor$ in a block whose size is $m \times n$ by modifying one pixel at most.

2.3 The Proposed Data Hiding Scheme for Binary Images

In this subsection, we shall first illustrate how to apply the proposed serial number matrix O to hide secret data in a binary image. Then, the extraction procedure will be presented.

A. The Embedding Procedure

Our proposed scheme not only uses a binary matrix K as the secret key but also uses a serial number matrix O to increase the number of the candidate modification positions. Therefore, the security of the embedded data is enhanced. The inputs to our scheme are as follows.

1) I is a host binary image (i.e., bitmap) to be modified to embed secret data. Here, I is partitioned into non-overlapped blocks B_i sized $m \times n$. For simplicity, we assume that the size of I is a multiple of $m \times n$.
2) K is a secret key shared between the sender and the receiver. It is a randomly selected bitmap sized $m \times n$.
3) r is the number of bits to be embedded in each $m \times n$ block of I, which is predetermined by the sender and receiver. The value of r satisfies $2^r - 1 \le mn$.
4) O is a serial number matrix shared between the sender and the receiver. It contains $2^r - 1$ non-duplicate integers at most. O_b is the binary representation of O.
5) S is critical information consisting of kr bits to be embedded in I, where k is the number of $m \times n$ blocks in I. S is divided into k groups, where each group consists of r secret bits. s_{ij} is the jth secret data embedded in the image block B_i. The order for each secret data in a group is from right to left. For example, assume the secret data is 0101 for block B_1, then their order is s_{11}=1, s_{12}=0, s_{13}=1 and s_{14}=0.

Assume that the size of K and O is 4×4. Let's consider a 4×4 image block B_i, which is a part of the host image I. The purpose is to show how to embed 4 (r=4) bits of data in B_i. Suppose we have the following inputs, shown in Fig. 2:

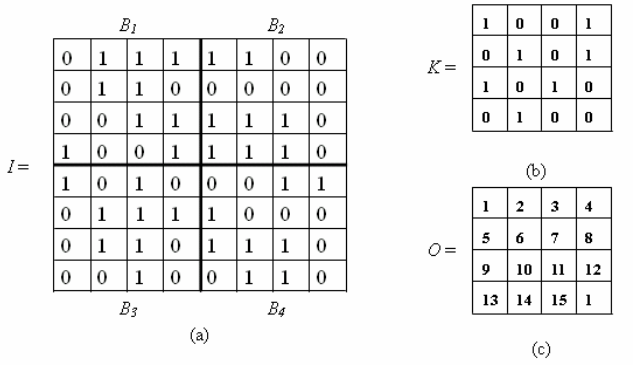

Fig. 2. (a) An example of binary cover image I sized 16×16, (b) A secret key K sized 4×4, and (c) A decimal serial number matrix O sized 4×4

The proposed embedding procedure takes the following six steps to process each block of cover image I.

Step 1. Compute $C_i = B_i \oplus K, 1 \le i \le k$.

Step 2. Generate r hiding equations according to Equation (1), and then obtain H_{ij} of C_i, where $1 \le j \le r$, and $1 \le i \le k$.

Step 3. Compute $H_{ij} \bmod 2$ of C_i to get relative h_i, where $1\ 1 \le j \le r$, and $1 \le i \le k$.

Step 4. Compare h_{ij} and s_{ij} of C_i. If they are identical, R_{ij} is set to be 0; otherwise, R_{ij} is set to be 1, where $1 \le j \le r$, and $1 \le i \le k$.

Step 5. Concatenate R_{ij}, where $1 \le j \le r$, to generate a R_i stream, and convert an R_i stream into the decimal representation as the modification position MP_i for B_i.

Step 6. Modify the pixel value of position MP_i of block B_i from 0 to 1 or 1 to 0.

Steps 1 to 6 are repeated until all blocks of the cover image I have been processed. At last, a stego-image I' is generated and is sent to the receiver. Then, to extract the hidden data, the secret key K and serial number matrix O are sent to the receiver in advance through a secure channel. A simple example is presented as follows using the cover image I, secret key K, and serial number matrix O shown in Fig.2. Assume that the secret data is 0010 0000 1011 0110. We can obtain I' shown in Fig. 3 after performing $B_i \oplus K$, $1 \le i \le 4$.

I' =

$C_1=B_1\oplus K$				$C_2=B_2\oplus K$			
1	1	1	0	0	1	0	1
0	0	1	1	0	1	0	1
1	0	0	1	0	1	0	0
1	1	0	1	1	0	1	0
0	0	1	1	1	0	1	0
0	0	1	0	1	1	0	1
1	1	0	0	0	1	0	0
0	1	1	0	0	0	1	0
$C_3=B_3\oplus K$				$C_4=B_4\oplus K$			

Fig. 3. The result of computing $B_i \oplus K$

Then, we can generate four hiding equations of block C_1 according to Equation (1) as follows. The remaining blocks also need to generate their data hiding equations to obtain H_{i1}, H_{i2}, H_{i3}, H_{i4}, where i=2, 3, and 4.

H_{11}=p$_1$(1)+p$_3$(3)+p$_5$(5)+p$_7$(7)+p$_9$(9)+p$_{11}$(11)+p$_{13}$(13)+p$_{15}$(15)+p$_{16}$(1).
H_{12}=p$_2$(2)+p$_3$(3)+p$_6$(6)+p$_7$(7)+p$_{10}$(10)+p$_{11}$(11)+p$_{14}$(14)+p$_{15}$(15).
H_{13}=p$_4$(4)+p$_5$(5)+p$_6$(6)+p$_7$(7)+p$_{12}$(12)+p$_{13}$(13)+p$_{14}$(14)+p$_{15}$(15).
H_{14}=p$_8$(8)+p$_9$(9)+p$_{10}$(10)+p$_{11}$(11)+p$_{12}$(12)+p$_{13}$(13)+p$_{14}$(14)+p$_{15}$(15).

$\bar{I}$ =

$\bar{B}_1$				$\bar{B}_2$			
0	1	1	1	1	1	0	0
0	1	1	0	0	0	0	0
0	**1**	1	1	1	1	1	0
1	0	0	1	1	1	1	0
1	0	1	0	0	0	1	1
0	1	1	1	1	0	0	0
1	1	1	0	1	**0**	1	0
0	0	1	0	0	1	1	0
$\bar{B}_3$				$\bar{B}_4$			

Fig. 4. A stego-image $\bar{I}$ generated by our proposed embedding procedure

After conducting Steps 3 to 6, we can generate the relative parameters of block C_i, shown in Table 3. From Table 3, we observe that MP_1 is 10 for block B_1, which means we only need to modify the pixel value of position 10 in block B_1 from 0 to 1 to hide secret data 0010 in block B_1. MP_2 is 0 for B_2, which means no pixel needs to be modified to embed secret data 0000. After modifying the pixel values in cover image I according to the modification positions (MPs) listed in Table 3, a stego-image $\bar{I}$ is generated as shown in Fig. 4.

B. The Extracting Procedure

In Fig. 4, $\bar{I}$ is the stego-image with the secret data already hidden in by the sender, and the receiver can only extract the secret data from it by using our proposed extracting procedure. Basically, the receiver has the same secret key K. S/he can generate $\bar{I}'$ (shown in Fig. 5) by computing $\bar{I} \oplus K$ and then extract the secret data by performing Steps 2 and 3 as described in our embedding procedure.

$\bar{I}'$ =

$\bar{C}_3=\bar{B}_3\oplus K$				$\bar{C}_3=\bar{B}_3\oplus K$			
1	1	1	0	0	1	0	1
0	0	1	1	0	1	0	1
1	1	0	1	0	1	0	0
1	1	0	1	1	0	1	0
0	0	1	1	1	0	1	0
0	0	1	0	1	1	0	1
0	1	0	0	0	0	0	0
0	1	1	0	0	0	1	0
$\bar{C}_3=\bar{B}_3\oplus K$				$\bar{C}_3=\bar{B}_3\oplus K$			

Fig. 5. An image $\bar{I}'$ which is result of performing $\bar{I} \oplus$K

To illustrate our extracting procedure, we use the above example to explain how to extract secret data from block $\bar{B}_1$ of stego-image $\bar{I}$ shown in Fig. 4. After receiving the stego-image $\bar{I}$, the receiver uses the shared secret key K to perform XOR computation on block $\bar{B}_1$ and generates the result $\bar{I}'$ shown in Fig. 5. Next, s/he constructs four data hiding equations according to the predetermined parameter r and the shared serial number matrix O, and learns that the values of H_{14}', H_{13}', H_{12}', *and* H_{11}' are 6, 4, 5, and 6, respectively. S/he can also obtain the values of h_{14}', h_{13}', h_{12}', and h_{11}', in this case 0, 0, 1, and 0, respectively, by performing H_{i1} mod 2, where i equals 1. At last, the receiver links the above four h_{i4}', where i equals 1, to get the extracted secret data related to $\bar{B}_1$ of the stego-image. The remaining secret data are extracted following the same procedure. The intermediate results and the extracted data for each block of the stego-image are listed in Table 4.

Table 4. The results of extracting data from stego-image $\bar{I}$

Block	$H_{i4}', H_{i3}', H_{i2}', H_{i1}'$	$h_{i4}', h_{i3}', h_{i2}', h_{i1}'$	Extracted Secret Data ($s_{i4}, s_{i3}, s_{i2}, s_{i1}$)
$\bar{C}_1$	6,4,5,6	0,0,1,0	0,0,1,0
$\bar{C}_2$	4,4,4,2	0,0,0,0	0,0,0,0
$\bar{C}_3$	3,4,5,3	1,0,1,1	1,0,1,1
$\bar{C}_4$	2,3,4,4	0,1,1,0	0,1,1,0

3 Security Analysis of the Proposed Scheme

In Fig. 1 (c), we select the first 15 positions in the serial number matrix O and assign increasing integers to them. For positions 1 and 16 of O, they are assigned the number "1", and their corresponding pixels in I' are the modification candidates when users need to modify H_{i1} from 0 to 1 or 1 to 0 to hide secret data. To enhance the security of the serial number matrix O and make the data hiding equations more complex and unpredictable, we can assign a different value, such as 10, to the 16th position of matrix O rather than 1. Assume that we give the serial number 10 to the 16th position, both H_{i2} and H_{i4} have to contain the value of the 16th position of matrix O, and H_{i1} will only contain eight rather than nine elements, according to the principle presented in Equation (1). Although the serial number matrix O has been modified, it still contains 15 non-duplicate integers, and therefore, we can conceal at least r secret data in a block and maintain very good stego-image quality. In addition, we can also design various versions of the serial number matrix O with different numbers of non-duplicate integers.

Our proposed serial number matrix O can have numerous variants. This characteristic offers the sender multiple modification positions, and thereby enhances the security of the hidden secret data. In general, if we want to embed r bits into an $m \times n$ block, the number of candidate modification positions is $C_{2^r-1}^{mn} \times (2^r-1)! \times (2^r)^{mn-(2^r-1)}$, where $C_{2^r-1}^{mn}(2^r-1)!$ means that we randomly select 2^r-1 elements out of the block and assign non-duplicate increasing integers to them. The remaining $m \times n - (2^r-1)$ positions of the block are faced with two possible cases, one is to be assigned with arbitrary values, which are identical to the values of other positions, and the other is not used for hiding secret data. Therefore, the value of each position has 2^r candidates. In [15], Tseng et al.'s scheme uses a weight matrix to represent the embedded data. The function of their weight matrix is like our serial number matrix O. Based on a weight matrix, the number of candidate modification positions is $C_{2^r-1}^{mn} \times (2^r-1)! \times (2^r-1)^{mn-(2^r-1)}$.

Please note that our proposed scheme allows users to appoint some positions of a block in the cover image as unchangeable positions those are not used to hide secret data. In contrast, in Tseng et al.'s scheme, each position of a block in the cover image is considered changeable and is used to hide secret data. Therefore, our hiding strategy is more complex, and provides better protection for the hidden data than Tseng et al.'s scheme does.

4 Experimental Results

In Tseng et al.'s scheme, they make two slight enhancements when they conduct their experiments to improve the image quality of the stego-image [15]. One is that pixels around black-and-white margins are modified at a higher priority, and the other is that no secret bits are concealed in an entirely black or white block. Their reason for taking those steps is that a block Bi,j may not be entirely black or white, but it could become completely black or white after some secret data are hidden in it. Their experimental results have confirmed that their enhancements are effective. Therefore, in our experiments, we also followed the same strategies to obtain good stego-image quality. Besides, we used three different types of images, all sized 512×512, as our host images, including an English text image, a Chinese text image, and the "Baboon" image, to hide the same amount of secret data as Tseng et al. did in their experiments (i.e. bits in an m×n block).

Table 6 presents the *PSNR* values of all the stego-images by our scheme and Tseng et al.'s scheme, respectively. As the results show, the *PSNR* values of our scheme are always higher than those of Tseng et al.'s scheme when the hiding capacities are the same. For both our new scheme and Tseng et al.'s scheme, the probability of occurrence of the case where *r* bits are embedded into a block without altering any pixel is $\frac{1}{2^r}$. Except for that case, our scheme always changes one bit at most in a block. By contrast, Tseng et al. may need to modify two bits in a block.

Table 6. Stego-image *PSNR*s generated by Tseng et al.'s scheme and our scheme

Block size / Host images	16×16		32×32	
	Our scheme	Tseng et al's scheme	Our scheme	Tseng et al's scheme
English text image	55.49 dB	53.51 dB	69.31 dB	67.08 dB
Chinese text image	55.60 dB	53.89 dB	69.31 dB	67.11 dB
Baboon	56.78 dB	54.69 dB	69.59 dB	67.40 dB

5 Conclusions

In this paper, we have offered a novel data hiding scheme to hide secret data in binary images. To provide a more complex hiding strategy than Tseng et al.'s, in our scheme, a shared key matrix and various versions of our proposed serial number matrix are used to decide in which candidate modification positions the $\lfloor \log_2(mn+1) \rfloor$ bits of secret data are to be hidden. We alter only one pixel value at most when hiding r secret data in an *m*×*n* image block, and the experimental results have confirmed that our stego-image quality is indeed better than that of Tseng et al.'s scheme. In the three cases in our experiments, the average PSNR value our scheme gave is greater than that given by Tseng et al.'s scheme. To sum up, our scheme enhances the security of the hidden secret data with simple operations; meanwhile, it also effectively improves the stego-image quality.

References

1. C. C. Chang, T. S. Chen and L. Z. Chung, "A Steganographic Method Based upon JPEG and Quantization Table Modification," *Information Sciences*, Vol. 141, pp.123-138 (2002).
2. H. Kobayashi, Y. Noguchi and H. Kiya, "A Method of Embedding Binary Data into JPEG Bitstreams," *IEICE Transactions*, Vol. J83-D2, No. 6, pp.1469-1476 (2000).
3. J. Fridrich, M. Goljan, and R. Du, "Detecting LSB Steganography in Color and Gray-scale Images," *IEEE Multimedia*, Vol. 8, Issue 4, Oct.-Dec. pp. 22-28 (2001).
4. M. Iwata, K. Miyake, and A. Shiozaki, "Digital Steganography Utilizing Features of JPEG Images," *IEICE Transactions on Fundamentals*, Vol. E87-A, No. 4, April, pp. 929-936 (2004).
5. C. H. Tzeng, Z. F. Yang, and W. H. Tsai. "Adaptive Data Hiding in Palette Image by Color Ordering and Mapping with Security Protection," *IEEE Transactions on Communications*, Vol. 52, No. 5, May, pp. 791- 800 (2004).
6. J. Chen, T. S. Chen, M. W. Cheng, "A New Data Hiding Scheme in Binary Image," in *Proc. Fifth Int. Symp. on Multimedia Software Engineering*. Proceedings, pp. 88-93 (2003).
7. J. Fridrich, "A New Steganographic Method for Palette-Based Images," in *Proc. of the IS&T PICS Conference*, Savannah, Georgia, April pp.285-289 (1998).
8. J. Spaulding, H. Noda, M. N. Shirazi and E. Kawaguchi, "BPCS Steganography Using EZW Lossy Compressed Images," *Pattern Recognition Letters*, Vol. 23, No. 13, pp.1579-1587 (2002).
9. J. Zhao and E. Koch, "Embedding Robust Labels into Images for Copyright Protection," in *Proc. Int. Conf. Intellectual Property Rights for Information Knowledge*, New Techniques, Munich, Germany, pp. 242-251 (1995).
10. K. L. Chung, C. H. Shen and L. C. Chang, "A Novel SVD- and VQ-based Image Hiding Scheme," *Pattern Recognition Letters*, Vol. 22, No. 9, pp.1051-1058 (2001).
11. M. Jo and H. D. Kim, "A Digital Image Watermarking Scheme Based on Vector Quantization," *IEICE Transactions on Information and Systems*, Vol. E85-D, No. 6, pp. 1054-1056 (2002).
12. M. Y. Wu and J. H. Lee, "A Novel Data Embedding Method for Two-color Facsimile Images," in *Proc. Int. Symp. on Multimedia Information Processing*, Chung-Li, Taiwan, R.O.C., Dec. (1998).
13. P. Tsai, Y. C. Hu and C. C. Chang, "An Image Hiding Technique Using Block Truncation Coding," in *Proc. of Pacific Rim Workshop on Digital Steganography*, Kitakyushu, Japan, July, pp. 54-64 (2002).
14. R. Z. Wang, C. F. Lin and J. C. Lin, "Image Hiding by Optimal LSB Substitution and Genetic Algorithm," *Pattern Recognition*, Vol. 34, No. 3, pp.671-683 (2001).
15. Y. C. Tseng, Y. Y. Chen, and H. K. Pan, "A Secure Data Hiding Scheme for Binary Images," *IEEE Transactions on Communications*, Vol. 50, No. 8, August, pp.1227-1231 (2002).
16. Y. K. Lee and L. H. Chen, "High Capacity Image Steganographic Model," in *Proc. of IEE International Conference on Vision, Image and Signal Processing*, Vol. 147, No. 3, pp. 288-294 (2000).
17. Min Wu and Bede Liu, "Data hiding in binary image for authentication and annotation," in *IEEE Transactions on Multimedia*, Vol. 6, Issue 4 , Aug., pp. 528 – 538, (2004).
18. Haiping Lu, Kot, A.C., and Jun Cheng, "Secure data hiding in binary document images for authentication," in *Proc. of the International Symposium on Circuits and Systems*, 2003(ISCAS '03.).Vol. 3 , 25-28 May, pp. III-806 - III-809 (2003).