Visual cryptography with extra ability of hiding confidential data

Wen-Pinn Fang
Ja-Chen Lin
National Chiao Tung University
Department of Computer and Information Science
1001 Ta Hsueh Road
Hsinchu, Taiwan
E-mail: jclin@cis.nctu.edu.tw

Abstract. *We present a two-in-one visual cryptography (VC) scheme that not only shares an image of moderate confidentiality between two noisy transparencies, but also hides in these two transparencies a more confidential text file describing the image. None of the transparencies alone can reveal anything about the image or text. Later, people can view the image by simply stacking the two transparencies; on the other hand, after certain simple computations, the more confidential text data can also be extracted. We also introduce an alternative version in which the decoding of both the image and text requires no computer. © 2006 SPIE and IS&T.* [DOI: 10.1117/1.2193912]

1 Introduction

Visual cryptography (VC) was a sharing technique first proposed by Naor and Shamir[1] in 1994. According to their design, VC can share an image (for example, "Lena") between two noisy transparencies (also known as shares). Then stacking these transparencies can unveil the input image directly using human vision without any computation. Many other VC systems[2–15] have been proposed since. On the other hand, there is another kind of tool to deal with sensitive data, namely, data hiding.[16–25] People can hide sensitive data in media; and later, the data can be extracted from these host media. Although there are many sophisticated VC systems (see Refs. 1–15), the reported works are seldom related to a VC design that, besides sharing to the input image ("Lena"), also hides a certain amount of data that are more private or more valuable than the input image itself (for example, Lena's address and other personal information). Therefore, we try to design in this paper a "two-in-one" VC scheme, which shares an image "Lena" of moderate confidentiality using two transparencies; and these two transparencies together can also extract a more confidential text data file describing the image "Lena." (More specifically, simply stacking the two transparencies can show the image "Lena" directly; meanwhile, after certain computations, one can also extract the more sensitive text data.) Of course, none of the transparencies alone can reveal anything about the image "Lena" or the more sensitive text.

Paper 05137R received Jul. 18, 2005; revised manuscript received Sep. 30, 2005; accepted for publication Oct. 25, 2005; published online Apr. 18, 2006.
1017-9909/2006/15(2)/023020/7/$22.00 © 2006 SPIE and IS&T.

There are several kinds of applications for this scheme. Perhaps the most direct application is to store or transmit an image of moderate confidentiality (for example, the photo of an employee, a suspect, or a VIP member) using the two generated transparencies, whose stacking together can yield the image. In addition, the background description of the image (for example, the employee's name, address, birthday, nationality, salary, and professional training) can be extracted only from joint information of the two transparencies by a high-ranking officer who knows the decoding key discussed in Sec. 4. As a result, if the two shares are collected, both of the lower rank officers and their commander can view the image; however, only the commander can decrypt the background text. Another application is to authenticate the transparencies generated by the VC system so that the combination containing faked transparencies can be identified, as discussed in Sec. 4.

The rest of this paper is organized as follows. The method is stated in Sec. 2; the experimental results are presented in Sec. 3; the application to the authentication of the transparencies is given in Sec. 4; the security concern is discussed in Sec. 5; an alternative version, whose decoding does not require a computer, is given in Sec. 6; finally, the conclusions are presented in Sec. 7. Throughout the paper, "Lena" is an input image of size 256×256, whereas "LENA" is the corresponding 512×512 stacking result obtained from stacking the two transparencies (both are 512×512) generated by the proposed method.

2 Method

We describe here how to create the two transparencies. Before doing so, we must introduce six fundamental blocks (type 0 through type 5, see Fig. 1) and the combinations to expand a white or a black pixel (see Figs. 2 and 3, respectively) of a given image "Lena."

The simplest design is to use 2×2 as the expansion rate, i.e., each share is 2×2 times greater than the input image "Lena" in size. Then, to define the fundamental blocks, we

Fig. 1 Six types of fundamental blocks (types 0 through 5).

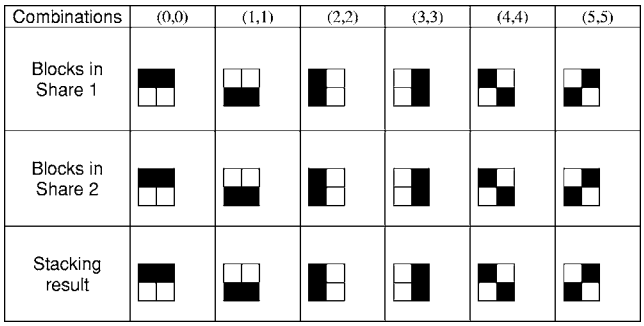

Fig. 2 Pairs whose stacking results represent white blocks.

consider all possible cases in which a 2×2 block has two white elements and two black elements. Therefore, we obtain $C(4,2)=4\times3/2=6$ types of fundamental blocks, called types 0, 1, 2, 3, 4, and 5, as shown in Fig. 1.

When we stack two shares and obtain the stacking result, the image is 2×2 times greater than the input image "Lena;" for each pixel of "Lena" is expanded to a 2×2 block of the stacking image "LENA." To maintain the black-and-white distribution (and hence the visual appearance) of the input image, the number of black elements in each black block of the stacking result "LENA" (a 2×2 block expanded from a black pixel of "Lena") should be more than that of each white block of the stacking result "LENA" (a 2×2 block expanded from a white pixel of "Lena"). For simplicity of the design, we assume that each black block has four black elements, and each white block has two white elements. (This obviously meets the rule that each black block has more black elements than each white block has.) Under this assumption, it is easy to see that if we want to share a white pixel of "Lena," we can use one of the six combinations {(0,0), (1,1), (2,2), (3,3), (4,4), (5,5)}, sketched in Fig. 2, to paint the corresponding position of the two shares. For example, the combination (3,3) means that, when we encode a white pixel, we may paint both shares 1 and 2 (at the block position corresponding to the white pixel being processed) using the fundamental block of type 3. Then in the decoding process, stacking these two shares will generate a white block (a 2×2 block having two white elements and two black elements) at the corresponding position. Analogously, we can use Fig. 3 to encode a black pixel of the input image "Lena."

Also note that to hide the more sensitive text file, we first transfer the text file to a sequence of digits of base 6. (Therefore, all digits are in the range 0 to 5.) Then this sequence is hidden.

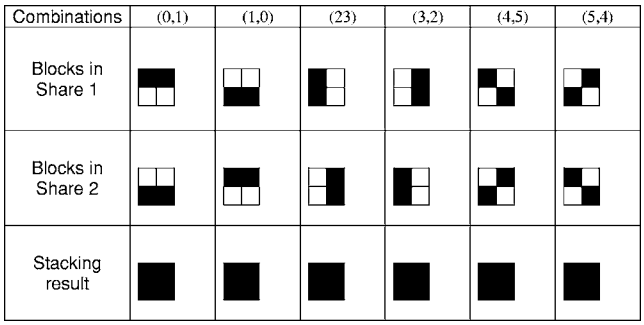

Fig. 3 Pairs whose stacking results represent black blocks.

Given next is the algorithm of the proposed method. The more sensitive text file is hidden first using part I of the algorithm; the input image "Lena" is then hidden using part II.

2.1 *Encoding Algorithm*

Part I. [To hide the confidential text data $\mathbf{d}=(d_1,d_2,d_3,\ldots,d_8,d_9,\ldots)$, where each d_i is a digit in the range 0 to 5.]

Step 1. Split each digit d_i into two parts x_i and y_i so that if a person only acquires one of the two numbers in $\{x_i,y_i\}$, he cannot extract d_i. More precisely, we randomly select a number x_i in the range 0 to 5, then find the corresponding y_i in the range 0 to 5 so that $(x_i+y_i)=d_i \bmod 6$. For example, if the confidential text data $\mathbf{d}=(d_1,d_2,d_3,\ldots,d_8,d_9,\ldots)=(1035004234\ldots)$, then we can decompose $\mathbf{d}$ as

$$1=3+4 \bmod 6,$$

$$0=1+5 \bmod 6,$$

$$3=1+2 \bmod 6,$$

$$5=0+5 \bmod 6,$$

$$0=2+4 \bmod 6,$$

$$0=0+0 \bmod 6,$$

etc.

Step 2. In an interleaved manner, the process sequentially generates half of the pixels in each of the two transparencies. For instance, $1=3+4 \bmod 6$, $0=1+5 \bmod 6$, and $3=1+2 \bmod 6$ in the preceding example, so the 2×2 fundamental blocks of types 3, 1, and 1 are the first, third, and fifth blocks of transparency 1, while the 2×2 fundamental blocks of types 4, 5, and 2 are the second, fourth, and sixth blocks of transparency 2. As for the remaining blocks, they are determined in Part II to share the image "Lena." In other words, as shown in Fig. 4, right now we have

Transparency 1: 3 ? 1 ? 1 ? 0 ? 2 ? 0 ? …,

Transparency 2: ? 4 ? 5 ? 2 ? 5 ? 4 ? 0…

Part II. (To share the input image "Lena")

Step 3. To determine the block types of the remaining blocks (the question marks written at the end of step 2) of the two transparencies, we sequentially read the pixel values of the input

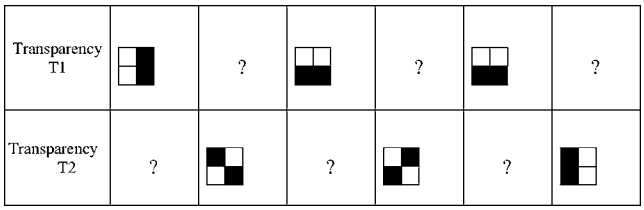

Fig. 4 Hiding the confidential text file (see steps 1 and 2 of the encoding algorithm).

image "Lena." Without the loss of generality, assume that, according to the scanning order, the pixel values are BBWBWB…. Since the desired visual decoding property requires that, after stacking, the 2×2 blocks had better also appear in the sequence BBWBWB…, thus the block types of the "?" can be determined by looking up the table in Fig. 2 (or Fig. 3) if a "W" (or a "B") is the current pixel value of "Lena." This results in the two transparencies shown in Fig. 5; in other words, we now have

Transparency 1: 3 $\underline{5}$ 1 $\underline{4}$ 1 $\underline{3}$ … ,

Transparency 2: $\underline{2}$ 4 $\underline{1}$ 5 $\underline{1}$ 2 … ,

Expected stacking: B B W B W B … .

This ends the algorithm.

In the decoding phase, directly stacking the two transparencies can reveal the (enlarged) image "LENA" without any computation, while the hidden text file $\mathbf{d}=(d_1,d_2,d_3,\ldots,d_8,d_9,\ldots)=1035004234\ldots$ can be extracted by the formula $(x_i+y_i)=d_i \bmod 6$ from the two transparencies. Here, x_i is the block type of the $(2i-1)$'th block of transparency 1, and y_i is the block type of the $(2i)$'th block of transparency 2.

3 Experimental Results

The experimental results are shown in Fig. 6. The size of each transparency is 512×512, which is 2×2 times larger than the 256×256 input image "Lena." The transparencies are noise-like. If we stack them, then we can get the visible image directly, as shown in Fig. 6(d). The two transparencies can also cooperate together to extract a sequence of 128×256 digits (each digit is in the range 0 to 5) hidden earlier, and this sequence is the confidential text for the purpose of validity verification or background description. The contrast (defined in Ref. 9) of our stacked result [Fig. 6(d)] is always $(4\text{-}2)/(2\times2)4=50\%$ because each 2×2 block of our stacked result always has four black elements if it is a black block, and two black elements if it is a white block. Notably, according to Ref. 9, the contrast is good if it reaches 50%. Therefore, the visual quality of our stacked result is not worse than those obtained using reported VC methods, while we have the extra advantage of being able to hide the confidential text describing the unveiled image.

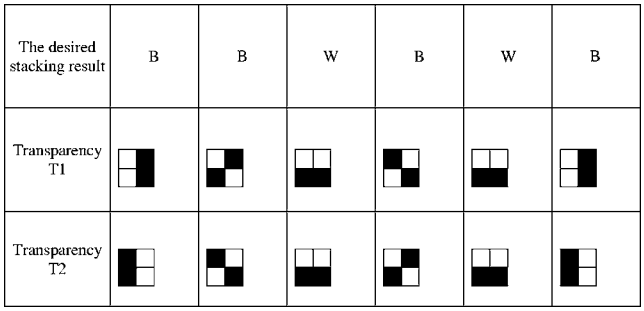

Fig. 5 Sharing the secret image "Lena" (see step 3 of the encoding algorithm) by filling in the undetermined parts of Fig. 4.

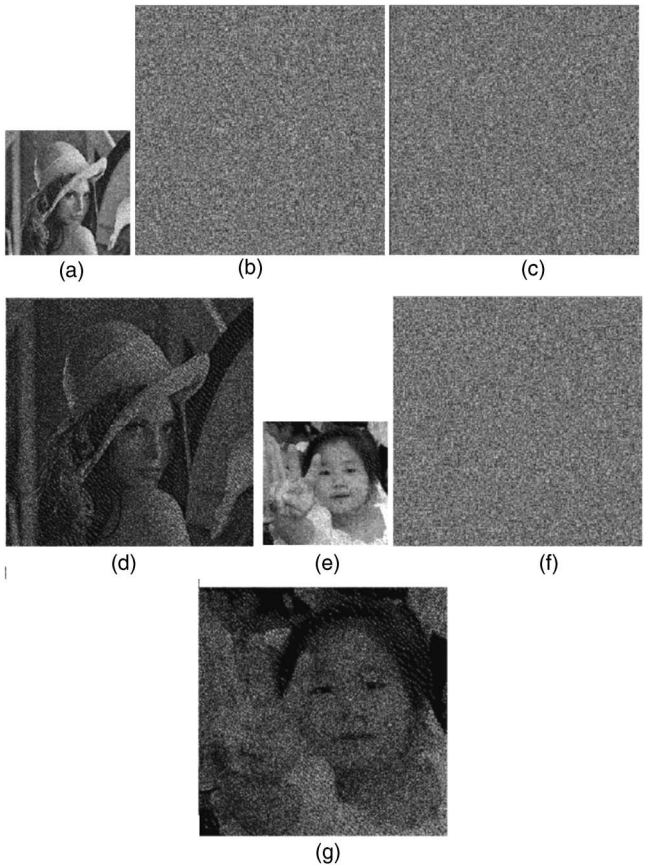

Fig. 6 Example illustrating the proposed method: (a) original halftone secret image "Lena," (b) and (c) two transparencies generated by our method, (d) image "LENA" obtained by stacking (b) and (c) [note that (b) and (c) can also be used to extract the hidden confidential text], (e) faked halftone image "Girl" owned by a hacker, (f) faked transparency, (g) faked image "GIRL" obtained by stacking (b) and (f). [However, by extracting the hidden text from (b) and (f), our ally can know that (g) is faked.]

4 Application to Transparency Authentication

As mentioned in Sec. 1, one of the applications of the proposed scheme is to authenticate the transparencies so that a combination of faked transparencies can be identified. This is explained in the following. We only have to transform an authentication message to a sequence **d** of digits in the range 0 to 5, and then hide this sequence according to part I of the algorithm to obtain shares 1 and 2. Now, assume that a hacker intercepts share 1 [Fig. 6(b)]. Using share 1

and his own faked input image "Girl" [Fig. 6(e)], the hacker may create a faked share [Fig. 6(f)], called share 2′, so that stacking shares 1 and 2′ will yield the faked stacking result "GIRL" [Fig. 6(g)]. However, the hacker cannot fool our agent/ally who is waiting at the receiver end of our network, since the faked pair {share 1, share 2′} cannot extract **d** (and hence, cannot extract our authentication message).

5 Techniques to Improve Security

Because shares 1 and 2 are transmitted using two distinct channels or stored in two different places, the chance that both shares are intercepted is very low. However, to reduce the damage that might be caused by a superhacker (or an internal betrayer of a company) who might intercept or access both shares 1 and 2, the security department can modify our algorithm by using their own manner to reassign the locations of the $\{x_i\}$ in share 1 and $\{y_i\}$ in share 2. Note that $\{x_i\}$ are not necessarily in the odd positions of share 1, and $\{y_i\}$ are not necessarily in the even positions of share 2 (see the final sentence of step 2 of the algorithm). For example, to paint the two shares (also called transparencies), the sequence $(x_1,y_1,x_2,y_2,x_3,y_3,x_4,y_4,x_5,y_5,\ldots)$ can be permuted by a random-position generator using a security seed. For instance, it becomes $(x_1,x_2,x_3,y_3,y_2,y_4,x_7,y_1,y_5,x_4,x_8,x_9,\ldots)$ after the permutation. Then the last two lines of step 2 become

Transparency 1: $x_1\ x_2\ x_3\ ?\ ?\ ?\ x_7\ ?\ ?\ x_4\ x_8\ x_9\ldots,$

Transparency 2: $?\ ?\ ?\ y_3\ y_2\ y_4\ ?\ y_1\ y_5\ ?\ ?\ ?\ldots.$

In the hidden-message extraction phase, the inverse of the random position generator is applied, using the same seed to reverse the sequence $(x_1,x_2,x_3,y_3,y_2,y_4,x_7,y_1,y_5,x_4,x_8,x_9,\ldots)$ back to $(x_1,y_1,x_2,y_2,x_3,y_3,x_4,y_4,x_5,y_5,\ldots)$. Then, as usual, $(x_i+y_i)=d_i \bmod 6$ are utilized to generate $(d_1,d_2,d_3,\ldots)$. Because the superhacker does not know how the designer distributes the $\{x_i\}$ and $\{y_i\}$, two things can be ensured even if both shares 1 and 2 are intercepted. First, the superhacker cannot extract the confidential text **d**. Second, the superhacker cannot use the information grabbed from the two intercepted shares (shares 1 and 2) to produce two faked shares (shares 1′ and 2′), which can also pass our authentication test. [In addition, stacking them yields a faked image similar to the "GIRL" image in Fig. 6(g).] Of course, if the pixels of the input image "Lena" are also permuted using a random-position generator, then the image "LENA" itself is also free from being viewed by the superhacker intercepting both shares 1 and 2. But this operation (permuting pixels of "Lena") is seldom used unless the image "Lena" itself is also highly confidential, because this operation reduces the advantages of visual cryptography: the image "LENA" can no longer be viewed by simply stacking the shares. (Recall that VC is a tool that balances between security and the benefit of "without-a-computer visual decoding" of images. If we only permute the locations of $\{x_i,y_i\}$, the disclosure of the image "LENA" still does not require a computer; although the disclosure of the confidential text **d** will then require a computer.)

6 When a Decoding Computer Is not Available

In our approach, the decoding of the "Lena" image by VC does not require a computer, but the decoding of the confidential text **d** does. Therefore, the method can be used in an environment where the decoding computer is not always available. For example, consider a system in which a single decoding computer can serve 100 teams, one at a time. Then, each team can decode its own moderately confidential image immediately (and begin the discussion about the visually decoded image within the team immediately). However, each team leader must wait (until it is his turn) for the computer server to help him in decoding the confidential text.

In the following, as suggested by one of the reviewers, we discuss how to modify the design so that the two new transparencies (generated by the modified design) can decode both "Lena" and the confidential text without using a decoding computer.

Apparently, if the confidential text can also be visually decoded, then there is no need to use a decoding computer. In this kind of approach, we have two images to deal with: the "Lena" image and its background description image. We plan to create two transparencies T_1 and T_2 having the following two properties: (1) stacking T_1 with T_2 yields "Lena" and (2) after the stacking mentioned in 1, if we fix T_1, and then shift T_2 by u units horizontally and v units vertically, then the stacking result becomes the confidential text image. Notably, the higher ranking officer keeps the values of u and v. If the higher ranking officer believes that it is not easy to shift a transparency by u units horizontally and v units vertically in the decoding phase, then he can use an auxiliary "nonnoisy" transparency T_3 (prepared earlier) in which only the boundaries of the transparencies T_1 and T_2 are sketched.

An encoding algorithm for this modified approach is given next, followed by a descriptive example and an experiment. In the algorithm, the image C is either a confidential text image describing "Lena" or a logo image for authentication purpose. The coordinate (i,j) indicates a pixel at location (i,j) of the image "Lena" (or the image C), or equivalently, the 2×2 block at location (i,j) of transparency T_1 (or transparency T_2).

6.1 *The Modified Encoding Algorithm (No Decoding Computer Required)*

The input to the algorithm are the two natural numbers u and v (u is the horizontal shift amount, and v is the vertical shift amount), a 256×256 image "Lena," and a confidential text image C.

The output is two noise-like transparencies T_1 and T_2 (both 512×512), which are useful in visually decoding both "Lena" and C.

The steps are

Step 1. For each coordinate (i,j) in the "easier-to-construct" areas, namely, $\{(i,j):0\leqslant i<u,0\leqslant j\leqslant 255\}$ and $\{(i,j):u\leqslant i\leqslant 255,0\leqslant j<v\}$, do the following:

1a. Randomly assign one of the six fundamental block types to $T_1(i,j)$.

1b. If "Lena" (i,j) is a white pixel, then copy the block type of $T_1(i,j)$ to $T_2(i,j)$; otherwise, copy the complement of the block type of $T_1(i,j)$ to $T_2(i,j)$.

Step 2. (For the remaining area where construction is more complicated):

2a. Initially, let the value of the counter k be 0.
2b. For each (i,j) satisfying $u\times k \leq i < u\times(k+1)$ and $0 \leq j < 255-v$, do the following:

2b.1. If $C(i,j)$ is a white pixel, then copy the block type of $T_2(i,j)$ to $T_1(i+u,j+v)$; otherwise, copy the complement of the block type of $T_2(i,j)$ to $T_1(i+u,j+v)$.
2b.2. If "Lena" $(i+u,j+v)$ is a white pixel, then copy the block type of $T_1(i+u,j+v)$ to $T_2(i+u,j+v)$; otherwise, copy the complement of the block type of $T_1(i+u,j+v)$ to $T_2(i+u,j+v)$.

2c.

2c.1. Add 1 to the value of k.
2c.2. If $k<(256/u)-1$, then go to step 2b; otherwise stop the algorithm. This ends the algorithm.

To help the readers understand step 2, we provide an example next. In the example, assume that $T_1 \ddagger T_2$ means stacking T_1 with T_2, i.e., "$\ddagger$" is the stacking operator.

6.2 *Example*

To make the explanation easier, we assume that the horizontal shift amount is $u=3$, and the vertical shift amount is $v=0$. Without the loss of generality, assume that the pixel values of the input image "Lena" and confidential text image C are

"Lena": BWBWBBBWWWBW…

Image C: WWBWBWBWB…

Since the horizontal shift amount is $u=3$, we split "Lena" and C into sectors of 3 pixels each. Therefore, we have

"Lena": BWB WBB BWW WBW…

Image C: WWB WBW BWB… .

Iteration 0a. Initially, because $u=3$, we randomly assign the block types to the three blocks of the first sector of T_1; for example, assign (0,4,3). So we have,

Transparency T_1: 0 4 3 ? ? ? ? ? ? ? ? ? ? ? ? …,

Transparency T_2: ? ? ? ? ? ? ? ? ? ? ? ? ….

Iteration 0b. For the first three blocks of the stacking result $(T_1 \ddagger T_2)$ to be (B,W,B), which are the three blocks in the first sector of "Lena," the first sector of T_2 should be $(\tilde{0},4,\tilde{3})$ due to the fact that the first sector of T_1 is (0,4,3). Here, $\tilde{0}$ is the complement of the block type 0, i.e., $\tilde{0}=1$. (See Fig. 3 for explanation; note that stacking block type 0 with block type 1 yields a black block.) Similarly, $\tilde{1}=0$, $\tilde{2}=3$, $\tilde{3}=2$, $\tilde{4}=5$, and $\tilde{5}=4$. Thus, we now know the first sector of T_1, and the first sector of T_2, i.e.,

T_1: 0 4 3 ? ? ? ? ? ? ? ? ? …,
T_2: 1 4 2 ? ? ? ? ? ? ? ? ? ….

Iteration 1a. For the three blocks of the first sector in $(T_1 \ddagger T_2^{\text{shifted}})$ to be (W,W,B), which are the first three blocks of the confidential text image C, the three blocks of the second sector of T_1 should be of types (1,4,3). Again, block type 3 is the complement of block type 2 implies $(1,4,3) \ddagger (1,4,2)=(W,W,B)$. Therefore, we have

Transparency T_1: 0 4 3 1 4 3 ? ? ? ? ? ? …,
Transparency T_2: 1 4 2 ? ? ? ? ? ? ? ? ? ….

Iteration 1b. Since the second sector of "Lena" is (W,B,B), therefore we let the second sector of T_2 be $(1,\tilde{4},\tilde{3})=(1,5,2)$. So we have

Transparency T_1: 0 4 3 1 4 3 ? ? ? ? ? ? …,
Transparency T_2: 1 4 2 1 5 2 ? ? ? ? ? ? ….

Iteration 2a. Read the next sector, i.e., (W,B,W), of the confidential text image C. Then, for $(T_1 \ddagger T_2^{\text{shifted}})$ to be (W,B,W) at the current location, the seventh, eighth, and ninth blocks of T_1 should be (1,4,2) since $(1,\tilde{5},2)=(1,4,2)$. Now we have

Transparency T_1: 0 4 3 1 4 3 1 4 2 ? ? ? …,
Transparency T_2: 1 4 2 1 5 2 ? ? ? ? ? ? ….

Iteration 2b. Since the third sector of the input image "Lena" is (B,W,W), let the third sector of T_2 be $(\tilde{1},4,2)=(0,4,2)$. Thus, we have

Transparency T_1: 0 4 3 1 4 3 1 4 2 ? ? ? …,
Transparency T_2: 1 4 2 1 5 2 0 4 2 ? ? ? ….

Iteration 3a. Read next sector, i.e., (B,W,B), of the confidential text image C. Then, for $(T_1 \ddagger T_2^{\text{shifted}})$ to be (B,W,B) at the current location, the 10th, 11th, and 12th blocks of T_1 should be $(\tilde{0},4,\tilde{2})=(1,4,3)$. Therefore, we have

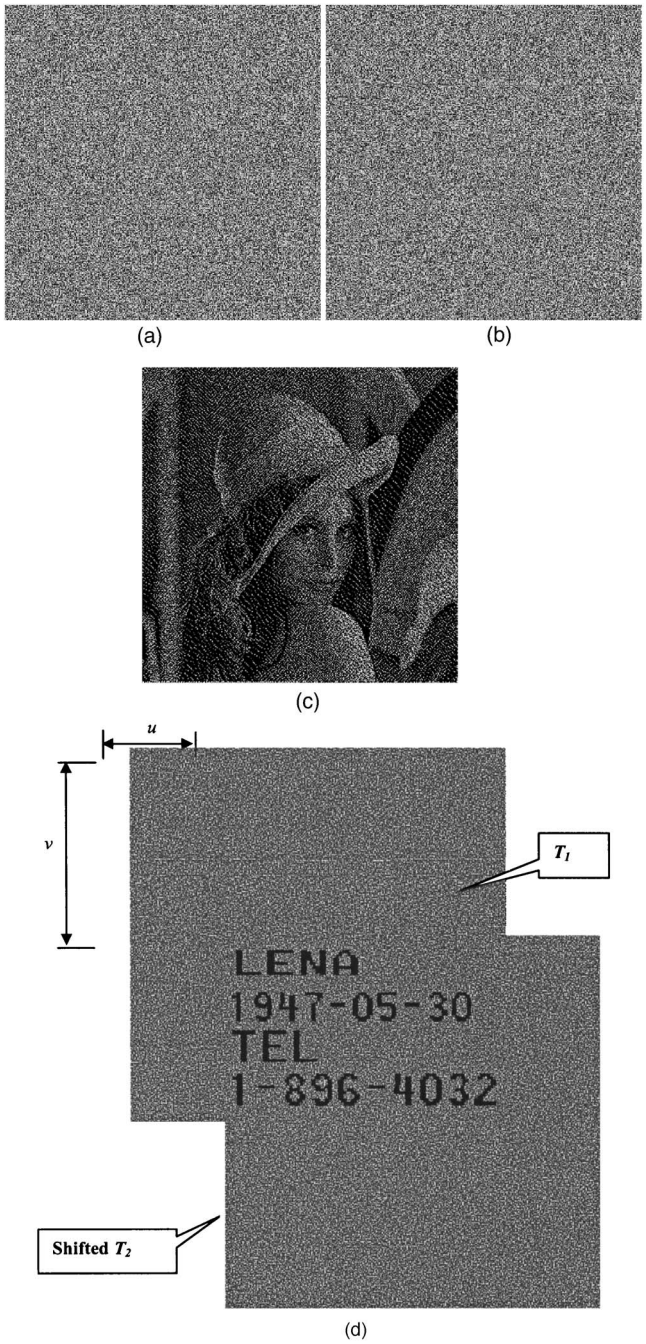

Fig. 7 Double-decoding without using a decoding-computer: (a) and (b) transparencies T_1 and T_2; (c) the result of stacking (a) and (b); (d) the result of stacking T_1 with "shifted T_2."

Transparency T_1: 0 4 3 1 4 3 1 4 2 $\underline{1\,4\,3}$ …,

Transparency T_2: 1 4 2 1 5 2 $\underline{0\,4\,2}$? ? ? ….

Iteration 3b. Since the fourth sector of the input image "Lena" is (W,B,W), the fourth sector of T_2 should be $(1,\widetilde{4},3)=(1,5,3)$. Thus, we have

Transparency T_1: 0 4 3 1 4 3 1 4 2 $\underline{1\,4\,3}$ …,

Transparency T_2: 1 4 2 1 5 2 0 4 2 $\underline{1\,5\,3}$ ….

Remaining iterations: These are similar to preceding iterations.

This ends the example.

An experiment using the modified encoding algorithm was performed and the results are shown in Fig. 7. The decoding process does not require the use of a computer; only the stacking operation is used to get Fig. 7(c) and 7(d).

7 Conclusions, Suggestions, and Future Work

We proposed a two-in-one method that not only visually shares an input image of moderate confidentiality, but also hides an extra text file that is more confidential. The extra hiding ability is achieved without giving any artificial appearance in the unveiled image. The unveiled "LENA" is lossless, in the sense that we can sequentially map from the 2×2 blocks of "LENA" back to the "exact" pixel values of "Lena." The method can also be applied to the authentication of the transparencies. We also discussed some other versions with improved security or easier decoding.

Notably, in the design introduced in Sec. 2, if the more confidential text is too long (for example, if "Lena's" personal information text requires more than $256\times256/2$ digits to express it, but the input image "Lena" is only 256×256), then we can enlarge the block size of the transparencies to solve the problem. [When we use 2×2 blocks (as we did so far), there are only six types of blocks. By using 3×3 blocks, there are many more block types that can be used.]

In future work, we will consider the following two topics. First, there is no computer available for decoding, we can still use visual decryption method to decrypt the confidential text. However, due to the natural limitations of visual resolution, the hidden text is much shorter (as compared with the data amount carried by the transparencies when a computer is available for decoding). Therefore, it will be interesting to discover a solution to increase the length of the text being carried, assuming that no computer can be used for decoding. Second, so far, our method deals with binary images. If we apply the method to the 8-bit planes of gray-value images, there might be some new applications.

Acknowledgments

The authors would like to thank the two reviewers whose comments have greatly improved this paper.

References

1. M. Naor and A. Shamir, "Visual cryptography," in *Cryptology-Eurocrypt'94, Lecture Notes in Computer Science*, Vol. 950, pp. 1–12 Springer-Verlag, Berlin (1995).
2. M. Naor and A. Shamir, "Improving the contrast via the cover base," in *Visual Cryptograpy II; Proc. Int Workshop on Security Protocols, Amale, Italy, April 1996*, M. Lomas Ed., pp. 197–202 (1996).
3. C. Blundo, A. De Santis, and D. R. Stinson, "On the contrast in visual cryptography schemes," *J. Cryptology* **12**, 261–289 (1996).
4. G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson, "Visual cryptography for general access structures," *Inf. Comput.* **129**, 86–106 (1996).
5. M. Naor and B. Pinkas, "Visual authentication and identification," in *Advances in Cryptology—CRYPTO'97*, B. Kaliski, Jr., Ed., Lecture

Notes in Computer Science, Vol. 1294, pp. 322–336, Springer, Berlin (1997).
6. D. R. Stinson, "An introduction to visual cryptography," presented at Public Key Solutions '97, Toronto, Canada (Apr. 1997); http://bibd.unl.edu/stinson/VKS-PKS.ps
7. A. Shamir, "Visual cryptanalysis," *Lect. Notes Comput. Sci.* **1403**, 201–210 (1998).
8. C. Blundo and A. de Santis, "Visual cryptography schemes with perfect reconstruction of black pixels," *Comput. Graph.* **22**, 449–455 (1998).
9. T. Hofmeister, M. Kruse, and H. U. Simon, "Contrast-optimal *k* out of *n* secret sharing scheme in visual cryptography," *Theor. Comput. Sci.* **240**, 471–485 (2000).
10. G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson, "Extended schemes for visual cryptography," *Theor. Comput. Sci.* **250**, 137–161 (2001).
11. C. C. Chang and J. C. Chuang, "An image intellectual property protection scheme for gray-level images using visual secret sharing strategy," *Pattern Recogn. Lett.* **23**, 931–941 (2002).
12. C. Blundo, A. D. Santis, and M. Naor, "Visual cryptography for gray level images," *Inf. Process. Lett.* **75**, 255–259 (2002).
13. Y. C. Hou, "Visual cryptography for color images," *Pattern Recogn.* **36**, 1619–1629 (2003).
14. C. C. Lin and W. H. Tsai, "Visual cryptography for gray-level images by dithering techniques," *Pattern Recogn. Lett.* **24**, 349–358 (2003).
15. R. Lukac and K. N. Plataniotis, "Bi-level based secret sharing for image encryption," *Pattern Recogn.* **38**, 767–772 (2005).
16. R. J. Anderson, "Information hiding: 1st international workshop," *Lect. Notes Comput. Sci.* **1174**, 317–333 (1996).
17. L. M. Marvel and C. T. Retter, "Hiding information in images," in *Proc. Int. Conf. on Image Processing* Vol. 2, pp. 396–398, Boston (1998).
18. L. M. Marvel, C. G. Boncelet, and C. T. Retter, "Spread spectrum image steganography," *IEEE Trans. Image Process.* **8**(8), 1075–1083 (1999).
19. T. Jamil, "Steganography: the art of hiding information in plain sight," *IEEE Potentials* **18**(1), 10–12 (1999).
20. H. H. Yu and P. Yin, "Multimedia data recovery using information hiding," in *Proc. IEEE GLOBECOM'00*, Vol. 3, pp. 1344–1348 (2000).
21. Y. C. Tseng, Y. Y. Chen, and H. K. Pan, "A secure data hiding scheme for binary images," *IEEE Trans. Commun.* **50**(8), 1227–1231 (2002).
22. C. C. Thien and J. C. Lin, "A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function," *Pattern Recogn.* **36**(12), 2875–2881 (2003).
23. M. Wu and B. Liu, "Data hiding in image and video. I. Fundamental issues and solutions," *IEEE Trans. Image Process.* **12**, 685–695 (2003).
24. C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recogn.* **37**(3), 469–474 (2004).
25. S. S. Maniccam and N. Bourbakis, "Lossless compression and information hiding in images," *Pattern Recogn.* **37**(3), 475–486 (2004).

Wen-Pinn Fang received his BS degree in mechanical engineering in 1994 from National Sun-Yet-Sen University and his MS degree in mechanical engineering in 1998 from National Chiao Tung University, where he is currently a PhD candidate in the Computer and Information Science Department. His recent research interests include pattern recognition and image processing.

Ja-Chen Lin received his BS degree in computer science in 1977 and his MS degree in applied mathematics in 1979, both from National Chiao Tung University, Taiwan, and his PhD degree in mathematics from Purdue University, U.S.A., in 1988. In 1981 to 1982 he was an instructor with the National Chiao Tung University and from 1984 to 1988 he was a graduate instructor with Purdue University. In August 1988 he joined the Department of Computer and Information Science at National Chiao Tung University, where he is currently a professor. His recent research interests include pattern recognition and image processing. Dr. Lin is a member of the Phi-Tau-Phi Scholastic Honor Society.