Color Visual Cryptography Scheme Using Meaningful Shares

Conference Paper · November 2008

DOI: 10.1109/ISDA.2008.130 · Source: DBLP

Hsien-Chu Wu
National Taichung University of Science and Technology

Color Visual Cryptography Scheme Using Meaningful Shares

Hsien-Chu Wu[1], Hao-Cheng Wang[2], and Rui-Wen Yu[3]

[1]*Department of Computer Science and Information Engineering, National Taichung Institute of Technology No. 129, Sec. 3, San-min Rd., Taichung City 404, Taiwan, R.O.C.*
[2]*Department of Computer Science and Engineering, National Chung Hsing University 250 Kuo-Kuang Road, Taichung, Taiwan 402, R.O.C.*
[3] *Graduate School of Computer Science and Information Technology, National Taichung Institute of Technology No. 129, Sec. 3, San-min Rd., Taichung City 404, Taiwan, R.O.C.*
wuhc@ntit.edu.tw, wanghc.tw@gmail.com, s18963102@ntit.edu.tw

Abstract

Visual cryptography (VC) schemes hide the secret image into two or more images which are called shares. The secret image can be recovered simply by stacking the shares together without any complex computation involved. The shares are very safe because separately they reveal nothing about the secret image. In this paper, a color visual cryptography scheme producing meaningful shares is proposed. These meaningful shares will not arouse the attention of hackers. The proposed scheme utilizes the halftone technique, cover coding table and secret coding table to generate two meaningful shares. The secret image can be decrypted by stacking the two meaningful shares together. Experimental results have demonstrated that the new scheme is perfectly applicable and achieves a high security level.

Keywords: *Visual secret sharing, visual cryptography, halftone technology*

1. Introduction

As a result of the astonishingly rapid advancement of various kinds of Internet technologies, more information are transmitted to all parts of the world from everywhere through the Net. Some of the objects transmitted online may be important secret images, and in such cases the senders have to take information security issues into consideration before they can trustingly enjoy the speed and convenience that nothing in this world but the Internet can offer.

In 1994, Naor and Shamir proposed a cryptography scheme called the “(k, n)-threshold visual secret sharing scheme,” and the idea they raised has ever since been referred to as “visual cryptography (VC)” [7]. The major feature of their scheme is that the secret image can be decrypted simply by the human visual system without having to resort to any complex computation. Naor and Shamir’s scheme can hide the secret image in n distinct images called shares. The secret image can be revealed by simply stacking together as many as k of the shares. Each of the shares looks like a collection of random pixels and of course appears meaningless by itself. Naturally, any single share, before being stacked up with the others, reveals nothing about the secret image. This way, the security level of the secret image when transmitted via the Internet can be effectively lifted up. Since Noar and Shamir published their VC scheme, many related methods have been developed and proposed [1-4, 8]. However, in addition to the meaningless shares they produce, those schemes take only binary images as secret images, which mean the contents of the secret images in most cases can be nothing but text or simple black-and-white designs. It is only natural now that researchers are more interested in developing new cryptography schemes that can also process secret color images [5, 9-12] that are more complex. Verheul *et al.* proposed a (k, n)-threshold color visual secret sharing scheme [11] based on pixel expansion for p-color images. Each pixel is expanded to p sections, and each section is divided into p subpixels and can produce n shares with p sections. When the k shares

978-0-7695-3382-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ISDA.2008.130

are stacked together, the *p*-color secret image is revealed. If *p* is large, then the pixel expansion is great too. Unfortunately, this scheme tends to produce many blocks with large numbers of black subpixels when revealing the secret image; in other words, the visual quality is a weakness. Besides, the shares are meaningless.

In order to reduce the expansion size of the secret pixel, Yang and Laih [12] proposed a *c*-color (*k*, *n*)-threshold visual secret sharing scheme. Their scheme can indeed reduce the number of black subpixels effectively. The pixel expansion of this scheme is $c \times m$, where *c* is the number of colors in the secret image, and *m* is the pixel expansion size of each color. At less pixel expansion, this scheme improves the visual quality of the revealed secret image. Afterwards, Shyu proposed an efficient *c*-color (*k*, *n*)-threshold visual secret sharing scheme [10] and has further improved the pixel expansion to maintain good visual quality of the revealed secret image. However, in spite of all the advancements both schemes have made, the shares produced by [11] and [12] are meaningless.

Hou proposed another color VC scheme [5]. Based on the halftone technique and color decomposition, it decomposes the secret image into three colors *C*, *M* and *Y*. By manipulating the three color values, the color pixels in the secret image can be represented. However, similar to what happens in [9-12], the shares are meaningless.

The shares such schemes as those generated in [5, 9-12] are meaningless and look like random dots. With such appearance, they make easy targets for attackers to zero in; whether or not the secrets can be easily cracked open, the looks of the meaningless shares are already revealing the existence of secrets to attackers. To fill in this security gap, a new method is presented in this paper to produce meaningful shares by using the halftone technique and two coding tables. Besides the camouflage of the shares, the new proposed method also has a strong encryption/decryption system that offers a high security level.

2. Related Works

2.1. The Basic Visual Secret Sharing Scheme

Noar and Shamir proposed the first visual secret sharing scheme of them all in 1994 [7]. Instead of the traditional cryptographic methods that require complex computation, Noar and Shamir's scheme uses the human visual system to decrypt the secret image. Furthermore, the scheme they proposed is a (*k*, *n*)-threshold visual secret sharing scheme. In other words, this method generates as many as *n* meaningless images called shares out of the secret image, and to decode the secret image requires as many as *k*, where $k \leqq n$, or more shares printed out on transparencies and stacked together. Otherwise, there is no way the secret image can be revealed out of the shares.

In a (2, 2)-threshold visual secret sharing scheme, let the secret image be a binary image with size $N \times N$. To begin with, every pixel is extended into a 2×2 block, and each block is composed of two black pixels and two white pixels as Fig. 1 shows. By referring to a predefined coding table, a block can be produced by corresponding to a related pixel of the secret image. When all the secret pixels are done processed, that is the moment the two shares are generated. Stacking the two shares together, we can reveal the secret image. After the encoding process, the size of the shares becomes $2N \times 2N$. The following are the secret image pixel coding rules. First, the system randomly picks one block from the six shown in Fig. 1 to represent *Share* 1 block. Second, a pixel is found that conforms to the secret image, and then the coding rules in Table 1 will be compared so that a matching block of *Share* 2 can be generated. When all the pixels are done processed, there will be two $2N \times 2N$ shares. Finally, stacking the two shares together, we can reveal the secret image. As effective and ingenious as this scheme may be, however, it can only process one secret image at a time, and the secret image can only be either text or a simple black-and-white design

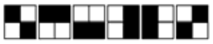
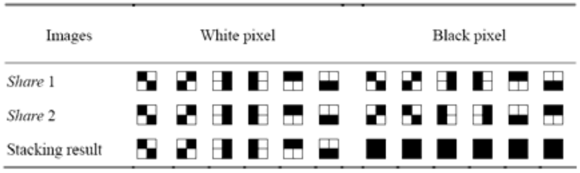


Fig. 1. Block group

Table 1. Coding Table of Share Blocks


Images
White pixel
Black pixel
Share 1
Share 2
Stacking result

2.2. Hou's Color Visual Secret Sharing Scheme

Hou [5] proposed three color VC methods where the same technique is used to decompose the color secret image into three separate images that are respectively colored cyan (*C*), magenta (*M*) and yellow (*Y*). Then the halftone technique is used to translate the three color images into halftone images. Finally, by combining the three halftone images, a color halftone image can be generated. The color halftone image generation process is shown in Fig. 2. The color halftone image takes eight different colors to display: cyan, magenta, yellow, black, red, green, blue and white. The three methods proposed take the color halftone image as the secret image. Here, we focus on the second method and describe the details of this method. For each pixel of the color halftone image, the following process must be done. First, 2×2 blocks are built according to *Share* 1, and the four pixels *C*, *M*, *Y* and *W* are randomly permuted. Then, the number of blocks is calculated for *Share* 2 according to the color ratio of the four pixels with the coding table (Table 3) referred to.

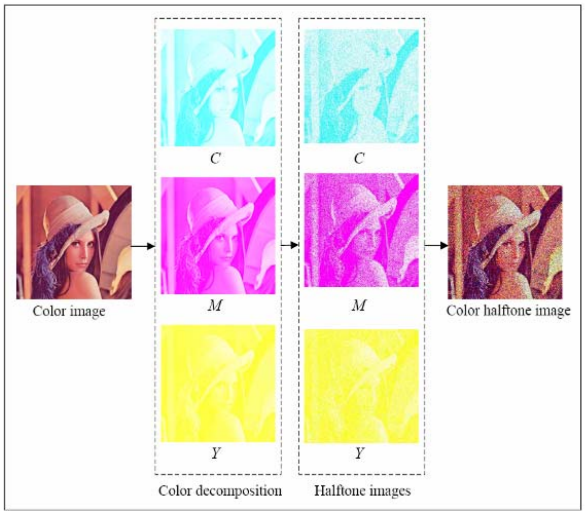


Fig. 2. Color decomposition

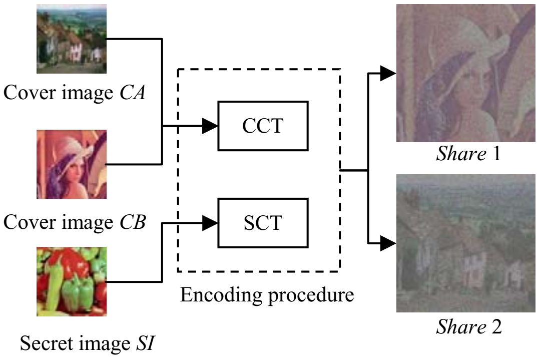
Table 3. Coding Table of Hou's Second Method (Secret Coding Table, SCT)

Shares \ Pixel	⬜	🩵	🟪	🟨	🟦	🟥	🟩	⬛
Share 1	🟨🩵 ⬜🟪	🟨🩵 ⬜🟪	🟨🩵 ⬜🟪	🟨🩵 ⬜🟪	🟨🩵 ⬜🟪	🟨🩵 ⬜🟪	🟨🩵 ⬜🟪	🟨🩵 ⬜🟪
Share 2	🟨🩵 ⬜🟪	🟨⬜ 🩵🟪	🟨🩵 🟪⬜	⬜🩵 🟨🟪	🟨🟪 ⬜🩵	🟪🩵 ⬜🟨	🩵🟨 ⬜🟪	🟪⬜ 🩵🟨
Stacked image	🟨🩵 ⬜🟪	🟨🩵 🩵🟪	🟨🩵 🟪🟪	🟨🩵 🟨🟪	🟨🟦 ⬜🟦	🟥🩵 ⬜🟥	🟩🟩 ⬜🟪	🟥🩵 🩵🟥

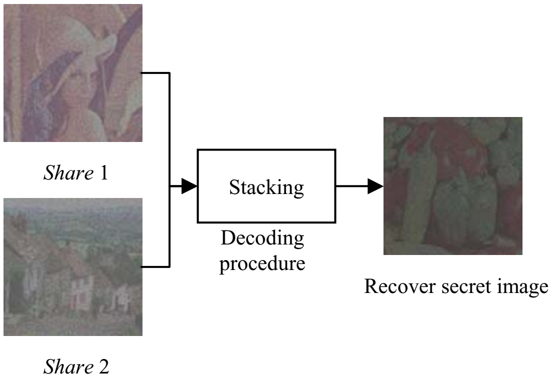
For example, if one pixel of the color halftone image is green, then the pixel's color ratio would be 100%, 0% and 100% for *C*, *M* and *Y*, respectively. Thus, block in *Share* 1 is the permutation of pixels: cyan, magenta, yellow and white. Then, the above information is applied, and the coding table will be referred to produce block of *Share* 2, where the permutation of the pixels is yellow, magenta, cyan and white. When all the pixels are done processed, two shares are produced. Each block of the two shares will be composed of *C*, *M*, *Y* and *W*. The secret image can be readily recognized visually when the two shares are stacked together.

3. The Proposed Scheme

There are four main procedures in the proposed scheme. The first procedure is color halftone transformation, where the color image is transformed to a color halftone image. The second procedure, pixel extraction process, extracts pixels from the color halftone image. Then, the following are encoding and decoding procedures, respectively. To generate the shares, two *N*×*N* cover images, named *CA* and *CB*, are used to encode the *N*×*N* secret image *SI* and make two 2*N*×2*N* shares called *Share* 1 and *Share* 2 (as shown in Fig. 3 (a)). *Share* 1 will be a meaningful share that appears just like *CA*, and *Share* 2 will be also a meaningful share that looks just like *CB*. Finally, during the decoding procedure, the secret image can be easily reconstructed by stacking *Share* 1 and *Share* 2 together as shown in Fig. 3 (b). There are two coding tables referred to in the encoding procedure: cover coding table (CCT) and secret coding table (SCT). As the names suggest, CCT is responsible for the encoding of the cover image, and SCT, on the other hand, is used to encode the secret image. The way SCT works in our new scheme is the same as it does in the second scheme of [5] (as shown in Table 3).



(a) Encoding procedure



(b) Decoding procedure

Fig. 3. Encoding and decoding procedures

3.1. Color Halftone Transformation and Pixel Extraction

Before encoding happens, this scheme applies color halftone transformation to produce color halftone images out of *CA*, *CB* and *SI*. Thus, *CA*, *CB* and *SI* are transformed into color halftone images *CA*', *CB*' and *SI*', respectively. The translation procedure is shown in Fig. 2. Next, the pixel extraction procedure is utilized for reducing the size of the color halftone image. The proposed scheme extracts some pixels from the color halftone image as important information for later coding. For each halftone image generated (as shown in Fig. 4), the pixels from the odd-numbered rows, or those from the even-numbered rows, can be extracted out to make the extracted image, which means the size of the extracted image is $N \times \frac{N}{2}$ as Fig. 5 shows. In such a way, *CA*', *CB*' and *SI*' are pixels extracted to generate *EA*, *EB* and *ES*. In other words, our new scheme can have the secret image restored with only half of the pixels at hand. This helps both save storage space in the main memory and shorten the encoding time.

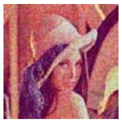


Fig. 4. Color halftone image

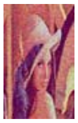


Fig. 5. Extracted image

3.2. Encoding and Decoding

During the encoding procedure, our new scheme takes in two coding tables, cover coding table (CCT) and the secret coding table (SCT), respectively. CCT is to help with the encoding of the extracted cover image, and SCT is to help process the extracted secret image. SCT in our new scheme works the same way as Table 3 does in the second method of [5].

In the encoding procedure, the proposed scheme uses CCT to encode *EA* and *EB*, while *ES* is encoded by the SCT. In CCT, as shown in Table 4, the first row represents various color pixels in *EA* and the first column stands for various color pixels in *EA*. The intersections of the rows and the columns are the output blocks with the left side of the block belonging to *Share* 1 and the right side of the block belonging to *Share* 2. SCT, as shown in Table 3, has the same definition as it does in [5]. In this paper, each pixel from the extracted image is expanded to one 2×2 block as shown in Fig. 6. The expanded block is placed in one of the 2×4 block patterns as shown in Fig. 7. By this way, the extracted image can produce a 2*N*×2*N* share.

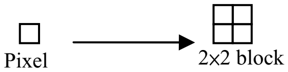


Fig. 6. Pixel expansion

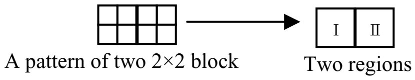
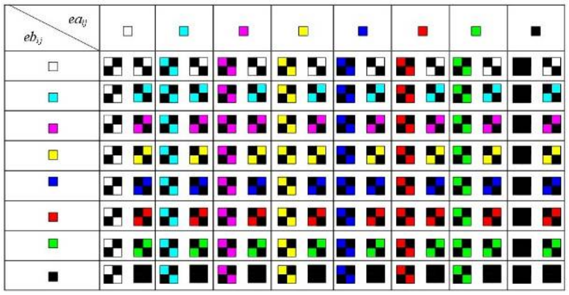
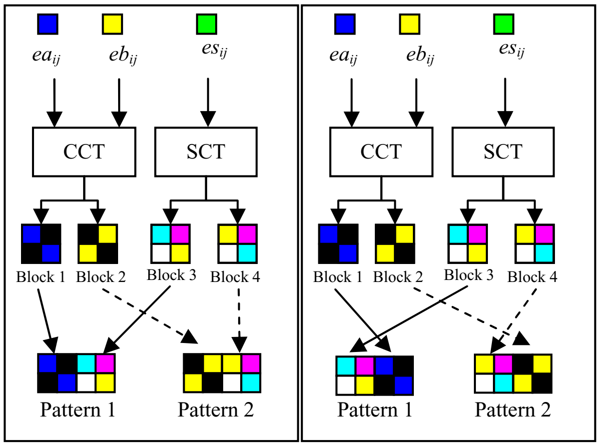


Fig. 7. Block division

After the color halftone transformation and pixel extraction, the proposed scheme has generated three color halftone images and extracted all the pixels it needs for *CA*, *CB* and *SI*, which are *EA*, *EB* and *ES*. As seen in Table 4, CCT is used to generate a 2×2 block from *EA*, and this block belongs to *Share* 1. As seen in Table 4 as well, CCT is used to generate a 2×2 block from *EB*, and this block belongs to *Share* 2. Then, the color ratio of the pixel is analyzed according to its position in the extracted image, for example ea_{ij} and eb_{ij} in *EA* and *EB*, where $0 \leqq i < N$ and $0 \leqq j < \frac{N}{2}$. According to the color ratio with Table 4 referred to, two 2x2 blocks, namely block 1 and block 2, can be produced. For example, if ea_{ij} and eb_{ij} are blue and yellow, then block 1 and block 2 are respectively and . The pattern is divided into two regions, region Ⅰ and region Ⅱ. Each region covers a 2x2 blocks area. Based on the position of pixel ea_{ij} or eb_{ij}, region Ⅰ or region Ⅱ is replaced with a suitable block.

Table 4. Cover Coding Table (CCT)





(a) The pixel belongs to odd row (b) The pixel belongs to even row

Fig. 8. An example of shares production

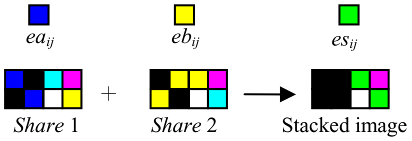


Fig. 9. An example of stacking *Share* 1 and *Share* 2

The replacement rules are as follows.

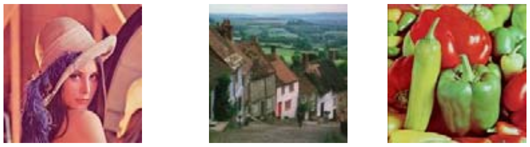
(1). When pixels ea_{ij} and eb_{ij} are on an odd row (i mod 2 = 0), replace region Ⅰ in pattern 1 with block 1, and replace region Ⅰ in pattern 2 with block 2. In contrast, if pixel ea_{ij} and eb_{ij} are on an even row (i mod 2 =1), replace region Ⅱ in pattern 1 with block 1, and replace region Ⅱ in pattern 2 with block 2. Now the encoding procedure for *EA* and *EA* is completed.

(2). For the encoding of *ES*, the proposed scheme needs to analyze the color ratio of the pixels. Then, according to the color ratio with the SCT (Table 3) referred to, block 3 and block 4 can be generated. The position es_{ij} in *ES* is defined, where $0 \leqq i < N$ and $0 \leqq j < \frac{N}{2}$. When pixel es_{ij} is on an odd row (i.e. i mod 2 =0), replace region Ⅱ in pattern 1 with block 3, and replace region Ⅱ in pattern 2 with block 4. In contrast, if pixel es_{ij} is on an even row (i.e. i mod 2 =1), replace region Ⅰ in pattern 1 with block 3, and replace region Ⅰ in pattern 2 with block 4. After completing pattern 1 and pattern 2, we put them in the matching positions in Share 1 and Share 2, respectively. When all the pixels of *EA*, *EB* and *ES* are done processed the production of *Share* 1 and *Share* 2 is completed. The generation rule for the pixels from the odd-numbered rows is shown in Fig. 8 (a), and the generation rule for the pixels from the even-numbered rows is shown in Fig.8 (b).

In the decryption process, we stack *Share* 1 and *Share* 2 together to reconstruct the secret image (see Fig. 9). Also, blocks representing ea_{ij} and eb_{ij} become black after the stacking, but will not affect the block which represents es_{ij}. Meanwhile, this can improve the contrast of the secret image and make the image clearer. The shares generation algorithm is as follows.

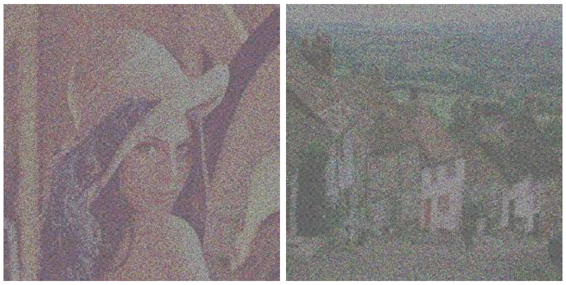
4. Experimental Results and Discussions

In these experiments, the secret image used was a 512×512 color image, and the cover images were also 512×512 color images. *Share* 1 and *Share* 2 were 1024×1024 pixels each. By stacking *Share* 1 and *Share* 2 together, the secret image "Peppers," as shown in Fig. 10 (c), can be revealed. The first cover image "Lena" and second cover image "Goldhill" are shown in Fig. 10 (a) and Fig. 10 (b), respectively. *Share* 1 and *Share* 2 are shown in Fig. 11 (a) and Fig. 11 (b), respectively. The reconstructed secret image is shown in Fig.12. As the experimental results have revealed, our new scheme can successfully conceal the secret image inside the meaningful shares, and later the secret image can be recovered simply by stacking *Share* 1 and *Share* 2 together. However, checking out the results in detail, we found that certain areas of the recovered secret image were darker in color than their counterparts in the original secret image. The cause can be either region Ⅱ or region I, depending on which one was black when *Share* 1 and *Share* 2 were stacked. As part of the experiments, we have also verified the security of the shares. Before producing block 3 and block 4, the proposed scheme must first learn the colors of the extracted pixels from the secret image. Then the obtained colors must meet their matches in the coding table so that a suitable block can be produced. The following combinations of a 2×2 block can be formed, for example, when a pixel is blue: ⊞and ⊞, ⊞ and ⊞, ⊞ and ⊞, ⊞ and ⊞, and so on. After all the coding table matching, a number of combinations will come out, and one can be randomly picked out to produce the block. The probability of the secret image block being guessed correctly can be calculated by the formula $\left(\frac{1}{p}\right)^{N\times\frac{N}{2}}$, where p is the number of block combinations there are, and $N\times\frac{N}{2}$ is the size of the extracted secret image. The probability of the extracted secret image being guessed correctly is extremely low. This way, with each block randomly produced, our new scheme makes it extremely difficult for an attacker to figure out what the secret image is.



(a)First cover image (b) Second cover image (c) Secret image

Fig. 10. Cover image and secret image



(a) *Share* 1 (b) *Share* 2

Fig. 11. Shares

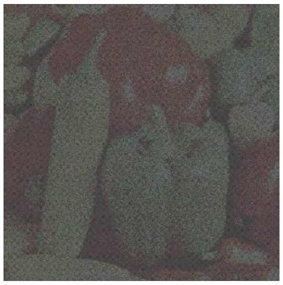


Fig. 12. Stacking *Share* 1 and *Share* 2

5. Conclusion

Currently, very few color VC schemes produce meaningful shares, but we consider this a pretty meaningful field of research to explore. In this paper, we offer a new color VC scheme we have developed that generates meaningful shares without increasing the security risks on the secret image. With this proposed scheme, we extend a single pixel into a 2×4 block. However, the size of the share remains the same as what happens in the 2×2 pixel expansion case. This way, a considerable part of the storage space can be saved, and more importantly, the shares do not look like random noise. In practical applications, our scheme can be combined with digital watermarking or visual verification systems.

References

[1] G. Ateniese, C. Blundo, A. De Santis and D. R. Stinson, "Visual cryptography for general access structures," *Information and Computation*, Vol. 129, pp. 86-106, 1996

[2] G. Ateniese, C. Blundo, A. De Santis and D. R. Stinson, "Contructions and bounds for visual cryptography," *Proceedings of the 23rd International Colloquium on Automata Languages and Programming*, pp. 416-428, 1996.

[3] C. Blundo and A. De Santis, "Visual cryptography schemes with perfect reconstruction of black pixels," *Comput. & Graphics*, Vol. 22, No. 4, pp. 449-455, 1998.

[4] S. Droste, "New results on visual cryptography," *Advances in cryptography: CRYPT'96*, Lecture Notes in Computer Science, No. 1109, Springer-Verlag, pp. 401-415, 1996.

[5] Y. C. Hou, "Visual cryptography for color images," *Pattern Recognition*, Vol. 36, pp.1619-1629, 2003.

[6] T. Katoh and H. Imai, "An extended constructions method of visual secret sharing scheme," *IEICE Trans. Fundamentals*, Vol. 179-A, No. 8, pp. 1344-1351, Aug. 1996.

[7] M. Naor and A. Shamir, "Visual cryptography," *Advances in Cryptology-EUROCRYPT'94*, pp. 1-12, 1995.

[8] M. Naor and A. Shamir, "Visual cryptography Ⅱ: improving the contrast via the cover base," *presented at Security in Communication Networks*, Sept. 1996.

[9] V. Rijmen and B. Preneel, "Efficient colour visual encryption for shared colors of benetton," *Eurocrypto'96*, Rump Session, Berlin, 1996.

[10] S. J. Shyu, "Efficient visual secret sharing scheme for color images," *Pattern Recognition*, Vol. 39, pp. 866-880, 2006.

[11] E.R. Verheul and H.C.A. van Tilborg, "Constructions and properties of *k* out of *n* visual secret sharing schemes," *Designs*, *Codes and Cryptography*, Vol. 11, No. 2, pp. 179-196, 1997.

[12] C. N. Yang and C. S. Laih, "New colored visual secret sharing schemes," *Designs*, *Codes and Cryptography*, Vol. 20, No. 3, pp. 325-335, 2000.