

Short Paper

A Note on Efficient Color Visual Encryption

CHING-NUNG YANG

*Department of Computer Science and Information Engineering
National Dong Hwa University,
Hualien, 974 Taiwan
E-mail: cnyang@mail.ndhu.edu.tw*

Rijmen and Preneel [1] proposed the colored VSS (Visual Secret Sharing) scheme, which only needs m sub pixels to represent $m!$ colors. However, the number of possible colors does not really reach $m!$ but only about $m!/2$ for large m . In this note, we will show this error and also give a correction for the Rijmen-Preneel scheme.

Keywords: visual cryptography, secret sharing scheme, visual secret sharing scheme, color visual cryptography, 2-out-of-2 VSS scheme

1. INTRODUCTION

A new type of secret sharing scheme [2] called VSS scheme, was first introduced by Naor and Shamir in 1994. The key features of the VSS scheme are that encodes a secret image is encoded into shadow images (called shares) and the decoder does not need knowledge of cryptography or cryptographic computations, but only the “eyes” of a human being. To decrypt the secret image, the reader should copy each share onto a separate transparency and then recover the secret image by stacking some of transparencies with the help of an overhead projector.

Some authors [1, 3-4] have proposed colored VSS schemes, where users can share a colored secret image. In [1], it was shown that the 2-out-of-2 colored VSS scheme only needs m sub pixels to represent $m!$ colors. However, the number of possible colors does not really reach $m!$. Herein, a correction for the Rijmen-Preneel scheme is given.

This note is organized as follows. In Section 2, we will give a brief review of the Rijmen-Preneel colored VSS scheme. Section 3 will show the correction.

2. RIJMEN-PRENEEL COLORED VSS SCHEME

The principle behind the Rijmen-Preneel scheme is that every pixel is divided into m sub pixels (m colors, where every sub pixel has a different color). By stacking two pixels, we will have $m!$ different permutations. The mixed color can be seen with the human eye, if the sub pixels are small enough. Finally, we can produce $m!$ colors. To produce these two shares, we first choose one permutation from $m!$ permutations to rep-

Received April 2, 2001; accepted December 19, 2001.
Communicated by Ja-Ling Wu.

represent the pixel in share 1 and then choose the corresponding permutation in share 2 such that the combination of the two pixels results in the desired color.

Example 1. For a 2-out-of-2 colored VSS scheme with six colors, represented as $0, 1, \dots, 5$, the six 2×6 colored matrices of the Rijmen-Preneel scheme are shown below.

Divide a pixel into three sub pixels with the colors c_0, c_1 , and c_2 . Since there are $3! = 6$ different permutations, so we can produce six colors, as shown in the following figure. (Note that when two different colors are stacked, we get a third color. For example: c_0 on c_1 gives c_3 , c_0 on c_2 gives c_4 , and c_1 on c_2 gives c_5 .) If the sub pixels are small enough, the “eyes” of a human being will average out the different possible combinations to different colors.

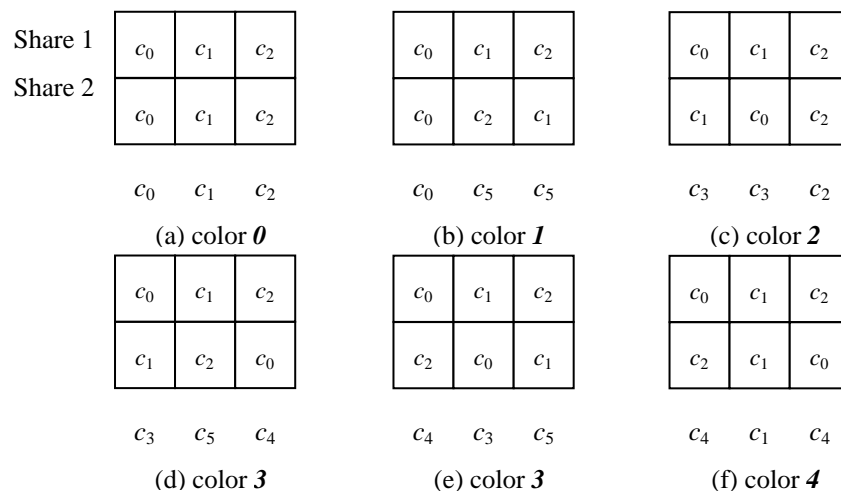


Fig. 1. Six different permutations in the 2-out-of-2 Rijmen-Preneel scheme.

The authors in [1] claimed that their scheme can produce 6 colors using three sub pixels. However, the number of colors does not really reach 6 for three sub pixels. In Fig. 1, it is shown combination (d) and (e) will produce the same color, 3 . Therefore, only five colors, $0, 1, 2, 3$, and 4 , can be produced by three sub pixels, not six. A detailed discussion and a correction for the Rijmen-Preneel scheme will be given in the next section.

3. DISCUSSION AND CORRECTION FOR RIJMEN-PRENEEL SCHEME

In [1], it was stated that “In the basic version of our scheme each pixel is divided into four sub pixels, with the colors *red*, *green*, *blue*, and *white*. These sub pixels can appear in any order. Taking symmetries into account, we get 24 different possibilities for the combination of two pixels. If the sub pixels are small enough, the human visual

system will average out the different possible combinations to 24 different colors.” However, this is not true because there are only $4! - 7 = 24 - 7 = 17$ different possible combinations as shown in Fig. 2. One can easily check these 24 combinations and find the following seven same combination pairs: (4, 5), (9, 13), (10, 19), (11, 14), (12, 20), (16, 21), (18, 23).

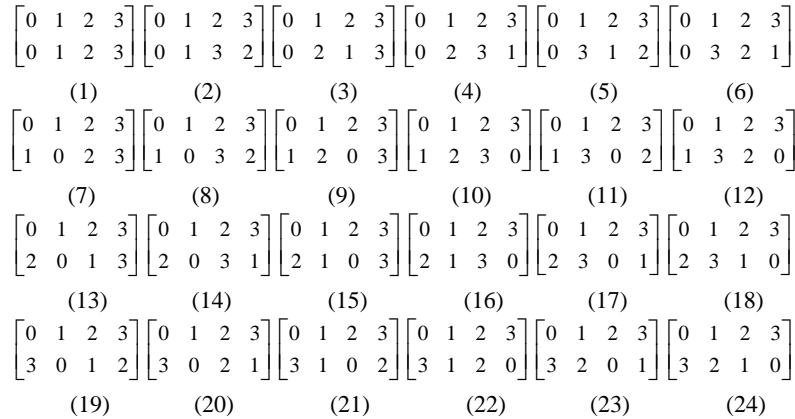


Fig. 2. 24 possible combinations of two pixels, where 0~3 denote red, green, blue, and white colors.

If a pixel is divided into five sub pixels, with five colors represented as 0 ~ 4, then there are $5! = 120$ combinations. Appendix shows 47 same combination pairs. Therefore, we have only $120 - 47 = 73$ different possibilities for combination of two pixels. Finally we produce 73 different colors.

A systematic procedure for finding the different combinations for different resolution m is described below.

A Systematic Procedure:

- (1) Use the notation $Comb(m, m)$ to represent the combination in which all m colors are stacked in the same position. Find the number $NUM_0 = Comb(m, m)$. In fact, it is obvious that $Comb(m, m) = 1$.
- (2) Find the combination in which all $(m - 2)$ colors are stacked in the same position, and then choose two other colors in the opposite pair, i.e., (i, j) and (j, i) . Calculate $NUM_1 = Comb(m, m - 2)$.
- (3) Find the combination in which all $(m - 2 \times n)$ colors are stacked in the same position, and then choose $2 \times n$ other colors in the opposite pair, where $n \geq 2$. Calculate $NUM_n = Comb(m, m - 2 \times n)$.
- (4) If $(m - 2 \times n) = 0$ (m : even number) or 1 (m : odd number), then stop.
- (5) Then, calculate the number of unique combinations in $m!$ combinations, $NUM_u = \sum_{i=0}^n NUM_i$.
- (6) The number of different combinations is $NUM_d = NUM_u + (m! - NUM_u)/2 = (m! + NUM_u)/2$.

Example 2. Find the number of different combinations for $m = 4$ and $m = 5$.

For $m = 4$, $NUM_0 = Comb(4, 4) = 1$, i.e., combination (1) shown in Fig. 2; $NUM_1 = Comb(4, 2) = 6$, i.e., combinations (2), (3), (6), (7), (15), and (22) shown in Fig. 2; $NUM_2 = Comb(4, 0) = 3$, i.e., combinations (8), (17), (24) shown in Fig. 2; therefore, $NUM_u = 1 + 6 + 3 = 10$. Finally $NUM_d = (m! + NUM_u)/2 = (4! + 10)/2 = 17$.

For $m = 5$, $NUM_0 = Comb(5, 5) = 1$, $NUM_1 = Comb(5, 3) = 10$, and $NUM_2 = Comb(5, 1) \times 3 = 15$, so $NUM_u = 1 + 10 + 15 = 26$. Finally, $NUM_d = (m! + NUM_u)/2 = (5! + 26)/2 = 73$.

4. TABLE AND CONCLUSION

Table 1 shows that the result in [1] is wrong, and the corrected result is given. From the last row in Table 1, we find that the percentage of reduction in the number of colors is about 50% for large m since $NUM_d/m! = 1/2 \times (m! + NUM_u)/m! \approx 50\%$.

Table 1. Correction for the 2-out-of-2 Rijmen-Preneel colored VSS.

resolution m	3	4	5	6	7	8	9	10
parameters								
$m!$ #1	6	24	120	720	5040	40320	362880	3628800
NUM_u #2	4	10	26	76	232	764	2620	9496
NUM_d #3	5	17	73	398	2636	20542	182750	1819148
Reduction (#3/#1×100%)	83%	71%	61%	55%	52%	51%	50%	50%

#1: original result in [1]

#2: unique combination in two pixels

#3: our corrected result $NUM_d = (m! + NUM_u)/2$

REFERENCES

1. V. Rijmen and B. Preneel, "Efficient colour visual encryption or 'Shared colors of benetton'," *Eurocrypt' 96 Rumpsession Talk*, <http://www.esat.kuleuven.ac.be/~rijmen/vc/>.
2. M. Naor and A. Shamir, "Visual cryptography," *Advances in Cryptology-EUROCRYPT' 94*, LNCS, Springer-Verlag, 1995, No. 950, pp. 1-12.
3. 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, 1997, pp. 179-196.
4. C. N. Yang and C. S. Lai, "New colored visual secret sharing schemes," *Designs, Codes and Cryptography*, Vol. 20, 2000, pp. 325-336.

$$\begin{aligned}
(31) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 0 & 3 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 2 & 0 & 4 & 1 \end{bmatrix} & (32) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 0 & 1 & 2 \end{bmatrix} \\
(33) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 0 & 2 & 1 \end{bmatrix} & (34) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 1 & 0 & 2 & 3 \end{bmatrix} \\
(35) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 1 & 0 & 3 & 2 \end{bmatrix} & (36) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 2 & 0 & 1 & 3 \end{bmatrix} \\
(37) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 2 & 0 & 3 & 1 \end{bmatrix} & (38) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 0 & 1 & 2 \end{bmatrix} \\
(39) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 0 & 2 & 1 \end{bmatrix} & (40) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 0 & 2 \end{bmatrix} \\
(41) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 4 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 0 & 3 \end{bmatrix} & (42) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 1 & 3 & 0 & 2 \end{bmatrix} \\
(43) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 0 & 3 \end{bmatrix} & (44) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 0 & 2 \end{bmatrix} \\
(45) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 2 & 3 & 0 & 1 \end{bmatrix} & (46) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 0 & 1 \end{bmatrix} \\
(47) \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 2 & 3 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 & 0 \end{bmatrix} & & &
\end{aligned}$$

Ching-Nung Yang (楊慶隆) 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 Ph.D. degree in Electrical Engineering from National Cheng Kung University in 1997. During 1987-1989 and 1990-1999, he worked at Telecommunication Lab., and Tainning Institute Kaohsiung Center, Chunghwa Telecom Co., Ltd., respectively. He is presently an assistant professor in the Department of Computer Science and Information Engineering at National Dong Hwa University. His research interests include coding theory, information security and cryptography.