A Visual Cryptography based system for sharing multiple secret images

Shang-Kuan Chen
Department of Computer Science and Information Engineering
Yuanpei University
Hsinchu, 300
Taiwan, R. O. C.

Abstract: - A Visual Cryptography (VC)-based system for sharing multiple secret images is proposed. Several weighted transparencies are generated so that people can reveal multiple secret images by stacking qualified subsets of some transparencies together. The transparency with relatively larger weight decides which secret image will be revealed. The proposed method has the following characteristics: decoding without computation, multiple secret images recovery, max-weight dominance, and quality-control design.

Key-Words: visual cryptography, image sharing, secret sharing, multi-secret sharing.

1 Introduction

Visual cryptography (VC) [1-4] is a kind of secret image sharing scheme that decodes a given secret image by human visual system without any computation. Naor and Shamir [1] introduced the so-called \((r, n)\)-threshold visual cryptography scheme. In their scheme, \(n\) transparencies, called shares, are generated. Stacking at least \(r\) of the \(n\) generated shares can decode the secret, but stacking less than \(r\) shares gives no information about the secret. In their encoding phase of VC, each pixel of the secret image is extended to a block for each of the \(n\) generated shares. Let \(p \times q\) be the block size. (Therefore, each share will be \(p \times q\) times bigger than the secret image in size.) In the stacking result, if at least \(d_b\) black elements exist in each block representing black, and if at most \(d_w\) black elements exist in each block representing white, then the contrast of the stacking result was defined in Ref. [2] as \(\frac{d_b - d_w}{p \times q}\).

Multi-secret images sharing system is an interesting research area for protecting many secret images. Wu and Chang [4] embedded two secret images into two circle shares. If two shares are stacked together, the content of the first secret image can be revealed. Moreover, rotating one of the two shares by a given angle degree, the other secret image can be revealed then. Their method is more flexibility than known VC methods. However, so far, only two secret images can be applied by their method.

Tsai et al. [5] and Feng et al. [6] proposed sharing methods for multiple secret images. In [5], they adopted XOR computing for embedding and extracting the secret images, and in [6], they adopted Lagrange’s interpolation that is applied from Thien and Lin [7] for generating several shares in order to recover the secret images. The method of [5] and [6] should use computer to extract the secret images. Although their methods are very convenient in network environment, but when in war, a computer may be not easy for use; the multi-secret image sharing must have other approaches, like VC.

Visual Cryptography for general access structure was introduced in [8-10]. Stacking each qualified subset of shares together can reveal the secret image, but stacking other, forbidden, sets of shares together has no information on the secret image. In [8-10], only single secret image is applied. Notably, in [6], their method can be applied on general access structure; however, as mentioned above, their method is not for VC, thus, can not reveal secret images by stacking the shares together.

In this paper, we propose a new VC scheme for progressive viewing, and the scheme shares \(k\) multi-secret images among \(k+1\) shares. The method is a kind of access structure based on VC. The forbidden set contains only subsets of which each contains single share. All other subsets, containing at least two shares, can reveal at least one of the secret images after stacking them together. The answer to the question “which secret image will be revealed?” is totally up to the share whose weight is relatively larger among the received shares.

The remaining parts of this paper are organized as follows: the proposed method is stated in Sec. 2; the experimental results are shown in Sec. 3; the concluding remarks are described in Sec. 4.
2 The proposed method

The proposed scheme shares $k$ binary secret images $S_1, S_2, \ldots, S_t$ to generate $k$ weighted shares and a basic weighted share. The $k$ shares $T_1, \ldots, T_t$ are with weights $w_1, w_2, \ldots, w_t$, respectively, in increasing order: $w_1 < w_2 < \cdots < w_t$. The basic share $T_0$ is with weight $w_0$, where $w_0 \leq w_t$; however, the best contrast will be achieved if $w_0 = w_t$. The reason will be stated in Sec. 2.1. Let $S$ consists of the binary pixels $\{b_{iz} | 1 \leq z \leq \text{size}(S_j)\}$. For each binary pixels $(b_{1z}, b_{2z}, \ldots, b_{tz})$ of the secret images $(S_1, S_2, \ldots, S_t)$, respectively, they are encoded into binary blocks $t_0, t_1, \ldots, t_k$ that belongs to $T_0, T_1, \ldots, T_k$, respectively. Let $p \times q$ be the number of binary terms in each block. The block $t_k$ is defined to have $w_k$ black terms and $p \times q - w_k$ white terms for each corresponding position $z$. In other words, each share will be $p \times q$ times bigger than the secret image in size. In this multi-secret VC system, the stacking result $T_0$ and some of $(T_j | j < i)$ together, the secret image $S_i$ is revealed. To reach the goal, the shares are designed in the following steps that will be stated in Sec. 2.2.

Before stating the encoding algorithm, the stacking result to represent the secret images will be defined in Sec. 2.1, as the number of black terms in each block of stacking results that represent each binary pixels $(b_{1z}, b_{2z}, \ldots, b_{tz})$ of the secret images $(S_1, S_2, \ldots, S_t)$.

2.1 The definition of stacking result

Let the stacking result to represent $S_i$, be $X_i$. For each block of $X_i$, the number of black terms corresponding to each original pixel $b_{iz}$ of $S_i$ is defined as follows.

1. **“White”** pixel of $S_i$ corresponds to $w_i$ black terms in the block $X_i$. Notably, the number of black terms defined to correspond to a “white” pixel of $S_i$ is the same for each $i$.

2. **“Black”** pixel of $S_i$ corresponds to the following two cases:

   (i $< k$): more than $w_i$ black terms and no more than $w_{i+1}$ black terms in the block $X_i$.

   (i = $k$): more than $w_i$ black terms and no more than $p \times q$ black terms in the block $X_i$.

For example, $(S_1, S_2)$ is the set of secret images, and $T_0, T_1, T_2$ are generated with weight $w_0, w_1$, and $w_2$. Therefore, for each position $z$, the number of black terms of blocks $t_0, t_1, t_2$ are $w_0, w_1, and w_2$, respectively. If the block of $X_i$ (the stacking result of $T_0$ and $T_i$) has $w_i$ black terms, the same as the number of black terms of $t_1$, then it corresponds to a “white” pixel of $S_1$. If the block of $X_i$ (the stacking result of $T_0$ and $T_i$) has more than $w_i$ but no more than $\min\{w_0 + w_1, w_2\}$ black terms of $t_1$, then it corresponds to a “black” pixel of $S_i$. Notably, if $w_0$ equals to $w_1$, the difference between the “white” and the “black” of $X_i$ will be maximized. Similarly, if the block of $X_2$ (the stacking result of $T_2$ and one or both of $T_0$ and $T_1$) has $w_2$ black terms, then it corresponds to a “white” pixel of $S_2$.

2.2 The encoding algorithm

Now, we state the encoding process. For every position $z$ in the corresponding block $t_0, t_1, \ldots, t_k$ of $T_0, T_1, \ldots, T_k$, respectively, the block $t_0, t_1, \ldots, t_k$ are generated by turns: the block $t_0$ of basic share $T_0$ is generated first, the corresponding block $t_1$ of $T_1$ is generated then, and finally the corresponding block $t_k$ of $T_k$ is generated. Let $u_i$ be an array to record the accumulative terms during the work of generating the corresponding blocks $t_0, t_1, \ldots, t_k$ of the shares $T_0, T_1, \ldots, T_k$.

Algorithm

For each position $z$ of the secret image, the steps are described as follows:

Step 1: (Generate basic share $T_0$)

1. For generating $t_0$, of basic share $T_0$, $w_0$ terms are randomly assigned to black and others are assigned to white.

2. After generating $t_0$, array $u_0$ records the black terms of $t_0$.

3. Set $i$ to 1.

Step 2: (Generate basic share $T_1$~$T_k$)

1. For generating $t_i$, of share $T_i$, there are two cases for concern:

   Case 1: the corresponding pixel of secret image $S_i$ is “white”.

   1. Let $g$ be a number of nonzero terms in current $u_i$. For these $g$ corresponding terms in $t_i$, they are assigned to black, and other terms in $t_i$ are assigned to white.

   2. The $w_i - g$ terms are randomly selected from zero-terms in $u_i$, and
for these corresponding terms in $t_z$, they are assigned to black. The rest unassigned terms in $t_z$ are assigned to white.

3. After assigning $t_z$, the corresponding terms of $u$ increase according to the $w_i$ black terms in $t_z$.

4. If $i < k$, then increase $i$ and return to Step 2.
   If $i = k$, then the work for this position is completed and go through next position.

Case 2: the corresponding pixel of secret image $S_i$ is “black”

Case 2.1: ($i < k$)

1. Select the minimal $w_i$ terms from the maximal $w_{i+1}$ terms in $u_z$.
2. For all corresponding terms of $t_z$, they are assigned to black.
3. After generating $t_z$, the corresponding terms of $u_z$ increase according to the black terms of $t_z$.
4. Increase $i$ and return to Step 2.

Case 2.2: ($i = k$)

1. Assign the minimal $w_i$ terms from $u$.
2. For all corresponding terms of $t_z$, they are assigned to black.
3. The work for this position is completed and go through next position.

The following example shows how to share three binary secret images $S_1$, $S_2$, and $S_3$. For each position $z$ and for each binary pixels $(b_{1z}, b_{2z}, b_{3z})$ taken from $(S_1, S_2, S_3)$, respectively, the binary blocks $t_{0z}$, $t_{1z}$, $t_{2z}$, and $t_{3z}$ are generated by turns. Let the weights $w_0 = w_1 = 3$, $w_2 = 5$, and $w_3 = 7$ for these shares and the size expansion of each share be $3 \times 3$.

The binary pixels $(b_{1z}, b_{2z}, b_{3z})$ must be in the following 8 cases: (W, W, W), (W, W, B), (W, B, W), (W, B, B), (B, W, W), (B, W, B), (B, B, W), and (B, B, B), where W means “white” and B means “black”.

Table 1 shows the generated blocks $t_{0z}$, $t_{1z}$, $t_{2z}$, and $t_{3z}$ for $(b_{1z}, b_{2z}, b_{3z})$ in these eight cases. Without the loss of generality, we show the process of generating the blocks $t_{0z}$, $t_{1z}$, $t_{2z}$, and $t_{3z}$ as follow, where $(b_{1z}, b_{2z}, b_{3z})$ are in the two cases (W, B, W) and (B, W, B).

In the case $(b_{1z}, b_{2z}, b_{3z}) = (W, B, W)$, $t_{0z}$ is first generated by randomly selecting $w_0 = 3$ black terms, as \( \begin{array}{c} \text{\includegraphics{black.png}} \end{array} \) and $u_z$ records accumulative terms as \{0, 1, 0, 0, 0, 1, 0, 0\}. Since $b_{1z} = W$, to generate $t_{1z}$ (with $w_1 = 3$), the black terms of $t_{0z}$ is followed.

Therefore, $t_{1z}$ is generated as \( \begin{array}{c} \text{\includegraphics{black.png}} \end{array} \). After $t_{1z}$ generated, $u_z$ records accumulative terms as \{0, 1, 2, 0, 0, 0, 0, 0\}. Since $b_{2z} = B$, to generate $t_{2z}$, the minimal $w_2 = 5$ terms are selected from the maximal $w_3 = 7$ terms of $u_z$. The 7 terms include three value-2-terms and four zero-terms. Therefore, only four zero-terms can be selected and additional one term should be randomly selected from value-2-terms of $u_z$, such that, $t_{2z}$ is generated as \( \begin{array}{c} \text{\includegraphics{black.png}} \end{array} \). Now, the stacking results are shown below: the stacking result of “$t_{0z}$ and $t_{1z}$ “ is \( \begin{array}{c} \text{\includegraphics{black.png}} \end{array} \), only $w_1 = 3$ black terms, thus it represents “white” in $S_1$; the stacking result of “$t_{1z}$ and $t_{2z}$ “ is \( \begin{array}{c} \text{\includegraphics{black.png}} \end{array} \), 7 black terms, representing “black” in $S_2$ (more than $w_2 = 5$ black terms); the stacking result of $t_{3z}$ and subset $(t_{0z}, t_{1z}, t_{2z})$ is \( \begin{array}{c} \text{\includegraphics{black.png}} \end{array} \), $w_3 = 7$ black terms, thus it represents “white” in $S_3$.

In the case $(b_{1z}, b_{2z}, b_{3z}) = (B, W, B)$, $t_{0z}$ is also generated by randomly selecting $w_0 = 1$ black term, \( \begin{array}{c} \text{\includegraphics{black.png}} \end{array} \), $w_1 = 1$ black terms, and $u_z$ records accumulative term as \{0, 0, 0, 0, 1, 0, 1, 0\}. Since $b_{1z} = B$, to generate $t_{1z}$, the minimal $w_1 = 3$ terms are selected from the maximal $w_2 = 5$ terms of $u_z$; thus, $w_2 - w_1 = 2$ terms are selected from zero-terms of $u_z$ and remaining one term are selected from value-1-terms of $u_z$. Therefore, $t_{1z}$ is generated as \( \begin{array}{c} \text{\includegraphics{black.png}} \end{array} \), and $u_z$ records accumulative terms as \{0, 1, 0, 2, 0, 1, 1, 0\}. Since $b_{2z} = W$, to generate $t_{2z}$, there should be $w_3 = 5$ black terms, and the number of non-zero terms of $u_z$ is equal to 5.
Therefore $t_2z$ is generated as $\text{[image]}$, and $u_z$ records accumulative terms as $\{0, 2, 0, 3, 0, 2, 2, 2, 0\}$. Finally, to generate $t_3z$, since $b_{3z}=B$, the minimal $w_3=7$ terms are selected from $u_z$. Therefore, four zero-terms of $u_z$ are selected, and then three terms are randomly selected from value-2-terms of $u_z$. The $t_3z$ is generated as $\text{[image]}

Table 1. Eight cases of encoding for each position $z$

<table>
<thead>
<tr>
<th>$b_{1z}$</th>
<th>$b_{2z}$</th>
<th>$b_{3z}$</th>
<th>$t_{0z}$</th>
<th>$t_{1z}$</th>
<th>$t_{2z}$</th>
<th>$t_{3z}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>W</td>
<td>W</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>B</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
</tr>
<tr>
<td>W</td>
<td>B</td>
<td>W</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
</tr>
<tr>
<td>W</td>
<td>B</td>
<td>B</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
</tr>
<tr>
<td>B</td>
<td>W</td>
<td>W</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
</tr>
<tr>
<td>B</td>
<td>W</td>
<td>B</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>W</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
<td>$\text{[image]}$</td>
</tr>
</tbody>
</table>

If we adequately assign the weights, quality-control design is also available. Obviously, when $T_0, T_1, \ldots, T_i$ are stacked together, the contrast is $w_{i+1} - w_i$ (if $i = k$, then $w_{i+1} = p \times q$). Therefore, if the value of $w_{i+1} - w_i$ is larger, then quality of...
stacking result is better. For example, there are two secret images \( S_1 \) and \( S_2 \) and the size expansion \( p \times q = 3 \times 3 = 9 \). If the stacking result representing \( S_1 \) (Lena) is not easier to be distinguish than that representing \( S_2 \) (YPU-logo), the weights can be assigned as \( w_0 = w_1 = 4 \), and \( w_2 = 8 \). When stacking \( T_0 \) and \( T_1 \) together, the contrast is
\[
\frac{w_2 - w_1}{p \times q} = \frac{8 - 4}{9} = \frac{4}{9}.
\]
However, when stacking \( T_0 \) \# 1, and \( T_2 \) together, the contrast is only
\[
\frac{p \times q - w_2}{p \times q} = \frac{9 - 8}{9} = \frac{1}{9}.
\]
If the stacking result representing \( S_2 \) (Lena) is not easier to be distinguished than that representing \( S_1 \) (YPU-logo), the weights can be assigned as \( w_0 = w_1 = 4 \), and \( w_2 = 5 \). When stacking \( T_0 \) and \( T_1 \) together, the contrast is only
\[
\frac{w_2 - w_1}{p \times q} = \frac{5 - 4}{9} = \frac{1}{9}.
\]
However, when stacking \( T_0 \) \# 1, \( T_1 \) and \( T_2 \) together, the contrast is
\[
\frac{p \times q - w_2}{p \times q} = \frac{9 - 5}{9} = \frac{4}{9}.
\]
Moreover, for the example of Table 1, the assignment of weights \( w_0 = w_1 = 3 \), \( w_2 = 5 \), and \( w_3 = 7 \) is an average contrast assignment for all secret images.

3 Experimental results
In the experiment, there are three secret images, shown in Fig. 1(a)-(c), are shared. The shares with weights \( w_0 = w_1 = 3 \), \( w_2 = 5 \), and \( w_3 = 7 \), shown in Fig. 2(a)-(d), respectively, are generated. Stacking Fig. 2(a) and (b) together, the secret message in Fig. 1(a) is revealed (see Fig. 3); stacking Fig. 2(b) and (c) together, the secret message in Fig. 1(b) is revealed (see Fig. 4); stacking Fig. 2(d) and at least one images from Fig. 2(a)-(c), the secret message in Fig. 1(c) is revealed. Fig. 5 shows the result of stacking all generated shares in Fig. 2.

![Figure 1](image1.png)
![Figure 2](image2.png)

Figure 1. The secret images.

![Figure 3](image3.png)

Figure 3. The result of stacking Fig. 2(a) and 2(b).
4 Concluding remarks

The proposed method is to share multiple secret images simultaneously. Several weighted shares are generated, and each is a binary noisy share. The revealing of multi-secret images is by stacking some shares together. Among the stacked shares, the one whose weight is relatively maximal will decide which secret image is revealed. The characteristics of the proposed method are (1) decoding without computation; (2) multiple secret image revealing; (3) the relatively maximal weight share will decide which secret image is revealed. Therefore, these properties make the tool applicable to the management of multiple secret images in a company. Another application is the so-called “game cards”. When a player begins to play the game, he can get game card $T_0$ initially. When he passes the test at stage $i$ ($i > 0$), he is granted the game card $T_i$. The player can view the secret image $S_i$ by stacking $T_i$ and $T_j$ together ($j$ is the number corresponding to an earlier stage).

References: