Overcoming Alignment Problem in Non-Identical Mathematical Support Visual Cryptography Schemes

Ari Moesriami Barmawi* & Widhian Bramantya

Graduate School of Informatics, School of Computing,
Telkom University, Jalan Telekomunikasi No.1, Bandung 40257, Indonesia
*E-mail: mbarmawi@melsa.net.id

Abstract. An important problem in visual cryptography is the alignment problem. Although Liu, *et al.* have proposed a method for aligning the shares, there is still a problem with the non-identical mathematical support visual cryptography schemes. For overcoming this problem, the Three-Orthogonal-Point (3OP) method is proposed in this paper. Based on the experimental result it was proven that it can overcome the alignment problem, while the time complexity for aligning the shares is decreased significantly from $O((m \times A)^2)$ to $O(m \times AI)$, for $AI < A$. The security is maintained, since an attacker cannot obtain any information related to the secret image.

Keyword: *alignment problem; decoding; encoding; orthogonal points; non-identical mathematical support; visual cryptography.*

1 Introduction

Several studies for the real-world implementation of visual cryptography have been conducted, such as the works of Hedge, *et al.* [1] and Hsu, *et al.* [2], which use visual cryptography for secure authentication in banking applications. An important problem in visual cryptography is the alignment problem. In 2009, Liu, *et al.* introduced a method for overcoming this problem [3]. They analyzed the structure of the basis matrices. Unfortunately, Liu, *et al.*'s method has a problem handling non-identical mathematical support visual cryptography schemes since there is no information about the exact position and orientation of each share. Therefore, alignment has to be done by brute force and consequently the time complexity is high. The Three-Orthogonal-Point (3OP) method is proposed in this paper for overcoming this problem.

2 Visual Cryptography Model

Visual cryptography is a cryptographic technique for encoding (encrypting) visual information. The information is decomposed into multiple shares, which can hence be decoded (decrypted) without using computation [4]. Suppose the secret information is in the form of an image. To encode the secret image (SI) it has to be decomposed into n shares, where $n \geq 2$. For decryption (recovering the

Received February 17th, 2017, Revised January 18th, 2018, Accepted for publication April 10th, 2018.
 ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2018.12.1.6

secret message), at least k shares need to be stacked to reveal the secret image, where $2 \leq k \leq n$. Construction of k out of n threshold access structures can be generalized. These are called general access structures and consist of the families of qualified sets and forbidden sets. A qualified set is a set of participants that can recover the secret image by stacking their shares, while a forbidden set is a set of participants that must not leak any information about the secret image. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be the set of n participants and $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ the set of shares. The secret image consists of black and white pixels.

2.1 Encoding and Decoding Process

The encoding process generates n shares from the secret image using the following process:

1. Construct $n \times m$ basis matrix M_0 (white) and M_1 (black), where n is the number of shares and m is the number of pixels. Each row of the matrix is denoted as m-vector V_i, where i should meet the following requirement: $1 \leq i \leq n$. The Hamming weight $H(V)$ of the OR-ed m-vector V_i is interpreted by the visual system as follows. For some fixed threshold and Hamming weight is h, where $1 < h < m$, a black pixel is interpreted if $H(V) \geq h$ and a white pixel is interpreted if $H(V) < (h - \alpha m)$, where the relative difference in weight is $\alpha > 0$. For example for Block Threshold VCS (BTVCS), the adopted threshold for determining whether a block of sub-pixels should be encrypted as a white or black pixel is $m/2$ [5].
2. Generating share collections for white pixels is done by permuting the columns of M_0. The group of these matrices is called C_0, where C_0 = {collection of matrices obtained by permuting the columns of matrix M_0}. A similar process must be executed on M_1 to generate the share collections for black pixels; the group of these matrices is called C_1, where C_1 = {collection of matrices obtained by permuting the columns of matrix M_1}. The share patterns generated from M_0 and M_1 are shown in Figure 1. If the shares are stacked and the sub-pixels are aligned correctly, then the secret image will appear.

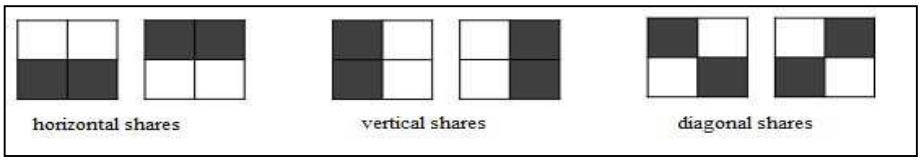
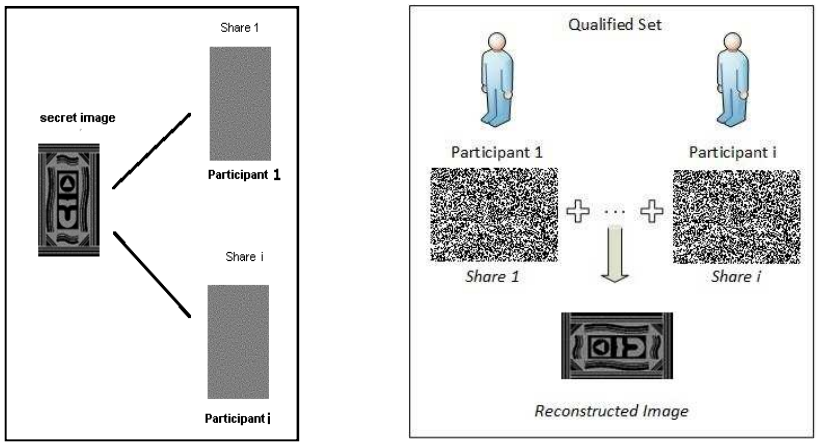


Figure 1 Various types of share patterns.

In visual cryptography, a solution is considered valid if the following three conditions are satisfied:

1. For any M_0 in C_0, the 'or' vector V of any k of m rows satisfies the Hamming weight $H(V) \leq l$ where l is equal to $h - \alpha \times m$ for fixed threshold $1 \leq l \leq m$ and relative difference $\alpha > 0$. l is the maximum number of black sub-pixels for each pixel such that it is categorized as a white pixel (representing the white threshold), and h is the minimum number of black sub-pixels for each pixel such that it is categorized as a black pixel (representing the black threshold).
2. For any M_1 in C_1, the 'or' vector V of any k of n rows satisfies Hamming weight $H(V) \geq h$.
3. For any subset of u rows $\{i_1, i_2, i_3,...,i_u\}$ of $\{1,2,...,n\}$ with $u < k$, the two collections of $u \times m$ matrices for $x \in \{0,1\}$, obtained by restricting each $n \times m$ matrix in C_t (where $t = 0, 1$) to rows $\{i_1,i_2, ...,i_u\}$, are indistinguishable in the sense that they contain the same matrices with the same frequencies.



(a) (b)

Figure 2 Encoding and decoding processes in VCS.

Based on [6,7], there is one participant P_1 who knows the secret image (SI). He/she decomposes the SI. He/she keeps share S_1 and distributes shares S_j to participants P_j for $2 \leq j \leq n$. Thus, each participant P_j holds a share S_j. The decoding process is conducted by stacking the shares. In this case, the participants in the qualified set meet and stack their shares into one. An illustration of the encoding and decoding processes is shown in Figures 2(a) and 2(b).

2.2 Non-Identical Mathematical Support Share

Most existing VCS methods use identical mathematical support shares. Identical mathematical support shares have the same shapes, sizes, orientations, and/or coordinate references. However, some VCS methods use different shapes, sizes, orientations, and/or coordinate references; they are called non-identical mathematical support VCS (NIMSVCS). The first NIMSVCS method was introduced by Machizaud, *et al.* [8]. NIMSVCS is used to increase the complexity of predicting the share structure. The encoding and decoding processes in NIMSVCS are similar to the encoding and decoding processes in IMSVCS. To avoid loss of generality, the shapes of the shares used in this study were restricted to squares and rectangles.

2.3 Alignment Problem

In general, the alignment process is conducted manually and is easy when the shares have small sub-pixel numbers and a large sub-pixel size. However, a large sub-pixel size will produce large shares, while shares with a small sub-pixel size are relatively hard to align [9]. The problem of aligning shares is called the alignment problem. To ease the alignment process, a larger size of shares is necessary. Thus, the alignment problem should be solved while maintaining the size of the shares.

3 Liu *et al.*'s Alignment Method

Due to high resolution as well as printing noise it is not easy to do precise alignment [3,10-12]. This problem also occurs in digital image shares. Several methods have been proposed for overcoming alignment problems [3,13,14]. One of the solutions has been proposed by Liu, *et al.* [3]. They modified the basis matrix such that the secret image can be recovered even though there has been a misalignment by one sub-pixel (shifted left or right). Based on Liu, *et al.*'s study [3], the structure of basis matrices M_0 and M_1 in (2,2)-VCS consists of four blocks. Basis matrices M_0 and M_1 are in the following forms (Eqs. (1) and (2)):

$$M0 = \begin{bmatrix} 1 \dots 1 & 0 \dots 0 & 1 \dots 1 & 0 \dots 0 \\ \underbrace{1 \dots 1}_{a} & \underbrace{0 \dots 0}_{b} & \underbrace{0 \dots 0}_{c} & \underbrace{1 \dots 1}_{d} \end{bmatrix} \quad (1)$$

$$M1 = \begin{bmatrix} 1 \dots 1 & 0 \dots 0 & 1 \dots 1 & 0 \dots 0 \\ \underbrace{1 \dots 1}_{a'} & \underbrace{0 \dots 0}_{b'} & \underbrace{0 \dots 0}_{c'} & \underbrace{1 \dots 1}_{d'} \end{bmatrix} \quad (2)$$

where *a, b, c, d, a', b', c', d'* are the sizes of the blocks (non-negative integers). These non-negative integers should satisfy $a + c + d = l$ and $a' + c' + d' = h$,

where l is the maximum number of black sub-pixels for each pixel such that it is categorized into white pixel (represented white threshold), and h is the minimum number of black sub-pixel for each pixel such that it is categorized as black pixel (represented black threshold). The contrast and security should satisfy the requirement as discussed in Section 2.1. If $e = b - b'$, the pixel expansion is equal to $m = a' + b' + c + d + 2e$. Liu *et al.*'s algorithm can reveal the secret image even if the shares are miss-aligned (shifted left or right) as shown in Figure 3. Suppose the number of shifted sub-pixels is r, then matrix M_0 is as shown in Eq. (3).

$$M0 = \begin{bmatrix} 1 \dots 1 & 0 \dots 0 & 1 \dots 1 & 0 \dots 0 & 0 \dots 0 & 0 \dots 0 & 0 \dots 0 \\ \underbrace{1 \dots 1}_{a'} & \underbrace{0 \dots 0}_{b'} & \underbrace{0 \dots 0}_{c} & \underbrace{1 \dots 1}_{d} & \underbrace{1 \dots 1}_{e} & \underbrace{1 \dots 1}_{e} & \underbrace{C1 \dots Cr}_{r} \end{bmatrix} \quad (3)$$

where C_1,..., C_r are the shifted pixels of share 2. Furthermore, the average contrast α_{avg} of the shifted scheme is as shown in Eq. (4) as follows [3]:

$$\alpha_{avg} = -\frac{(m-r)e}{m^2(m-1)} \quad (4)$$

α_{avg} reflects how clear the image can be perceived visually. Furthermore, after shifting r sub-pixels, the value of the average contrast is negative ($\alpha_{avg} < 0$) since the transparency size should be kept minimal. To reduce the transparency size, reducing the amount of pixel expansion and the size of each sub-pixel is necessary [15]. However, the smaller the share size, the more difficult it is to align the transparencies together.

Thus, the problem of Liu, *et al.*'s method is that for overcoming the alignment problem they need a larger transparency size. The difficulties increase as the number of shares in non-identical mathematical support form increases.

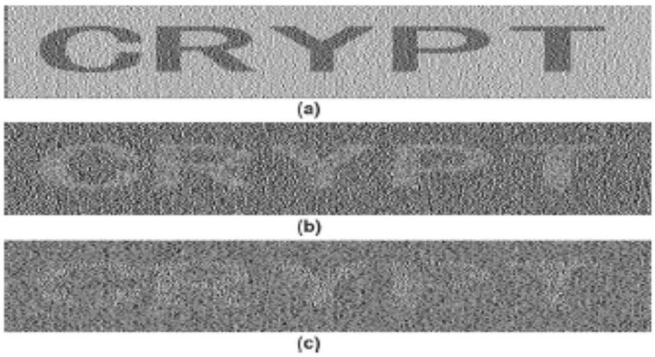


Figure 3 (a) Stacking result without shifting. (b) Stacking result with one sub-pixel shifting. (c) Stacking result with two sub-pixel shifting.

4 Alignment Method for Non Identical Mathematical Support Share (NIMS) Visual Cryptography

For decreasing the difficulty of aligning non-identical mathematical support shares, the 3-Orthogonal-Point (3OP) method is proposed in this paper. It is based on a three-point pair generated from two orthogonal vectors that are perpendicular with each other, such that the three-point pairs will form an angle of 90 degrees. As a result, during the stacking process the three pairs of points are easy to find and localize. Since the 3OP should be kept secret, a securing process is implemented using the secret sharing concept. Suppose a bank officer (P_1) wants to authenticate a customer (P_2), then the officer creates a secret message and generates two shares (encoding process). The second share is given to the customer and the first share is kept by the officer. When the customer tries to conduct a transaction, he/she should first be authenticated by the officer by stacking the customer's share and the officer's share (decoding process). When the secret image is revealed, the customer is authenticated.

In order to implement the secret sharing concept, visual cryptography with a Chameleon hash function [16] was used. The Chameleon hash function is proposed because it can make the alignment process faster and there is a possibility to regenerate the same 3OP but with different numbers embedded in the shares. Thus, it can be used for securing the 3OP by camouflaging the 3OP. In this case, each point of the 3OP is generated by two numbers (3OP parameters), where one number is embedded in the first share (S_1) and the other one in the second share (S_2). For generating the 3OP, the secret image size is required. After determining the secret image size, the process is continued with determining the shadow shares. A shadow share is a template that is used together with the basis matrix to determine the shares.

For the decoding process, two processes are conducted: calculating the 3OP and the stacking process. Calculating the 3OP is done by calculating the coordinates using the numbers embedded in the shares. After obtaining the coordinates of each point, both shares can be stacked and the secret image will be revealed.

4.1 Encoding Process

The encoding process consists of three sub-processes: pre-decomposing, decomposing, and embedding the 3OP parameters into the shares. The pre-decomposing process is a process for generating the 3OP based on the secret image size.

In the decomposing process, participant P_1 generates shares S_1 and S_2, and the parameters of the 3OP are embedded into the shares. The parameters of the 3OP are used to calculate the coordinates of the 3OP using the Chameleon hash

function. This function is used because its characteristics provide the ability to regenerate points in the same position using different parameters, such that it can be used to secure the 3OP. The details of the Chameleon hash function are discussed in Section 4.1.1. An overview of the pre-decomposing process is shown in Figure 4.

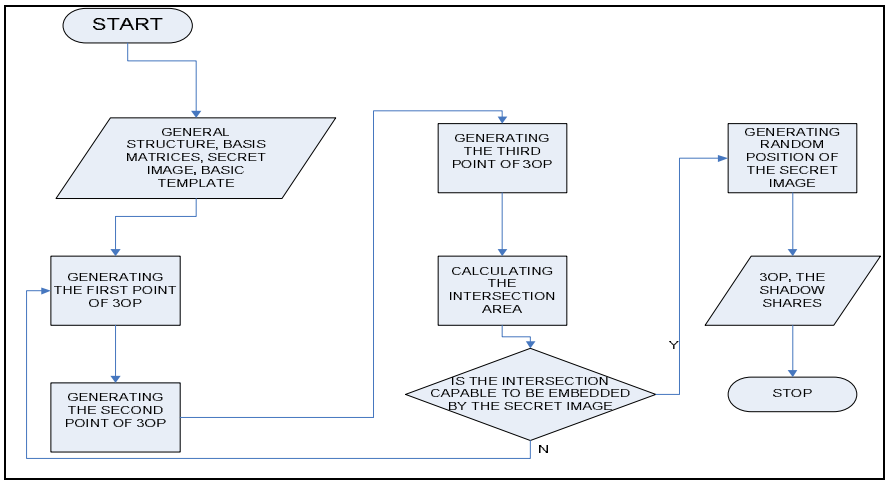


Figure 4 Overview of the pre-decomposing process.

4.1.1 Generating the 3OP

After determining the general access structure and the basis matrices, participant P_1 determines the secret image (SI), including its size as well as the sizes of all shadow shares (SS_i). For generating the 3OP, two prime numbers p and q are determined such that $p = kq + 1$, where q is a large prime factor. Furthermore, an element g of order q in Z_q^*, where Z_q^* is a group of non-zero integers that is closed for multiplication. P_1 chooses the private number $x \in Z_q^*$, then value $r \in Z_q^*$, and calculates the Chameleon hash function $y(m, r) = g^m h^r \bmod p$, where $h = g^x \bmod p$. In this case, different parameters m and r can generate the same value of y due to the collision concept of the modulo function. Finally, the 3OP is calculated by P_1 using the chameleon hash function for obtaining x_1 and y_1, x_2 and y_2 as well as x_3 and y_3 as the coordinates of the 3 points using Eqs. (5) and (6).

$$x_1 \equiv (((g^{\beta 1})(h^{\gamma 1}) \bmod p) \bmod (\text{the width of } SS\text{'s in pixels})) + 1 \quad (5)$$

$$y_1 \equiv (((g^{\beta 1})(h^{\gamma 1}) \bmod p) \bmod (\text{the height of } SS\text{'s in pixels})) + 1 \quad (6)$$

where x_1 is the x-ordinate of the first point; g is the generator of p with order q; β_1 and γ_1 are random elements of p with order q, and p is a prime number. A similar calculation is used for determining the coordinates of other points.

The 3OP should be aligned in the vertical and horizontal direction. Since there are still possibilities that the 3OP is not in the vertical and horizontal direction, a normalization process should be conducted. After the normalization process, the normalized 3OP of the shares should have the same orientation. For adjusting the orientation, the following process is conducted:

1. Pick the orthocenter of the 3OP from both shares and stack them.
2. Make sure that the three points of the first share (*S1*) and the second share (*S2*) are in the same position. For achieving this goal, *S2* can be rotated 90°, 180° and 270° until the proper position is found.
3. The 3OP is generated after the three points of the two shares are in the same position, such that the secret image is revealed.

4.1.2 Calculating Intersection Area

After generating the 3OP, participant P_1 calculates the intersection area. Calculating the intersection area is necessary since there is a possibility that the intersection area generated by the 3OP is smaller than the area of the secret shares. For calculating the intersection area, the scenario as shown in Figure 5 is used.

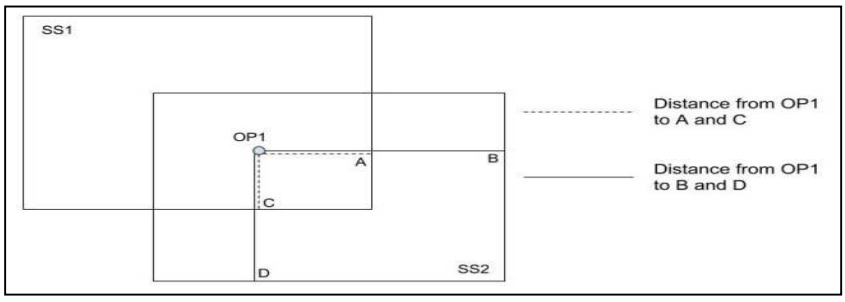


Figure 5 Scenario for calculating the intersection area.

Suppose the width of SS_1 and SS_2 are w_1 and w_2 respectively. OP_1 is the first point of the 3OP, OP_1 – A is the distance from OP_1 to the right border of share SS_1. The distance is equal to l_1. OP_1 – B is the distance from OP_1 to the right border of share SS_2, which is equal to l_2. If l_2 is greater than l_1, then the intersection's width is $W_{intr} = l_1 + (w_2 - l_2)$. However, if l_2 is smaller than l_1, then the intersection's width is $W_{intr} = l_2 + (w_1 - l_1)$.

For calculating the height of the intersection, the distance from OP_1 to the top border of the shares is necessary. In this case, suppose the heights of SS_1 and SS_2 are h_1 and h_2 respectively, then OP_1 – C is the distance between OP_1 and the bottom border of share SS_1, which is equal to s_1, OP_1 – D is the distance from OP_1 to the bottom border of share SS_2, which is equal to s_2. If s_2 is greater than

s_1, then the intersection's height is $H_{intr} = s_1 + (h_2 - s_2)$. However, if s_2 is smaller than s_1, then the intersection's height is $H_{intr} = s_2 + (h_1 - s_1)$. Thus, the intersection area is $W_{intr} \times H_{intr}$.

After calculating the intersection area, P_1 should check whether the intersection size can be embedded by the secret image. If it is not, then P_1 should repeat the 3OP generation process.

4.1.3 Decomposing Process

The decomposing process consists of two processes, dithering and share generation. Dithering is conducted if the secret image is in greyscale (8-bit representation). The objective of this process is to convert 8 bits into 1 bit. Based on [17], pixels $I(i,j)$ of continuous-tone digital image are processed in a linear fashion, left-to-right and top-to-bottom. The algorithm compares the grey-level intensity value of the current pixel with a threshold (128). If the input greyscale value is greater than the threshold, the pixel is considered black and its value $B(i,j)$ is 1, else the pixel is white and $B(i,j)$ is 0. The process is shown in detail in Figure 6. An error occurs if there is a difference between the original greyscale value and the threshold. To reduce the error, it is distributed to four unprocessed pixels as shown in Figure 7.

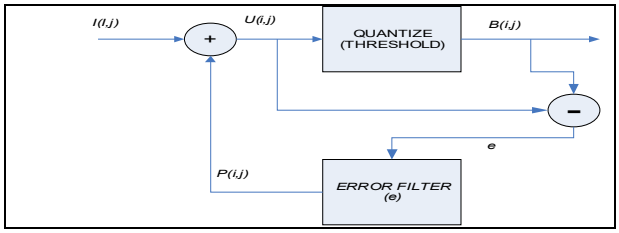


Figure 6 Floyd-Steinberg dithering process [17].

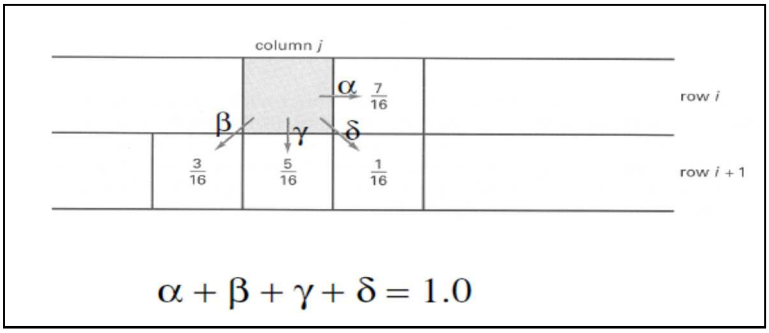


Figure 7 Error dispersed to neighboring pixels [17].

Figures 6 and 7 show that the modified pixel value after spread quantization can be calculated using Eq. (7).

$$U(i,j) = I(i,j) \sum_k \sum_l w(k,l) e((i-k),(j-l)) \quad (7)$$

where $e((i-k)(,j-l)) = U(i-k, j-l) - B(i-k, j-l)$ and w is the weight that is represented by $\alpha, \beta, \gamma, \delta$.

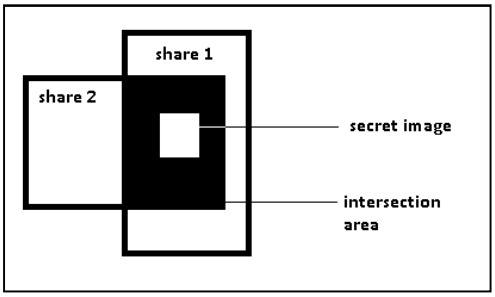


Figure 8 Structure of the stacked area.

For generating the shares, the two shadow shares should be stacked using the 3OP, as shown in Figure 8, where there are three areas: the intersection area, the area outside the intersection, and the area inside both the intersection and the secret image. In the share generation process, the two following rules are used:

1. For the intersection area inside the secret image, each pixel is expanded using the basis matrices based on the pixel value (e.g. if the pixel value is 0, then the share pixel should be expanded using the black basis matrix and vice versa).
2. For the area outside the intersection, all pixels should be expanded using the black basis matrix.
3. For the area inside the intersection, all pixels should be expanded using the white basis matrix, such that after stacking the shares it is easy to identify the secret image.

For securing the 3OP, after generating the shares the 3OP parameters should be modified such that an attacker cannot obtain the original 3OP, even though the modified 3OP is obtained. Modification of the 3OP is conducted by reducing the parameters β_1, β_2, β_3 and γ_1, γ_2, γ_3 by μ (μ should be agreed upon and kept secret by both parties), such that $\beta_1' = \beta_1 - \mu$, $\beta_2' = \beta_2 - \mu$, and $\beta_3' = \beta_3 - \mu$. A similar process is conducted to γ_1, γ_2, γ_3 for obtaining γ_1', γ_2', and γ_3'. The modified 3OP parameters g, p, β_1', β_2', β_3' should be embedded into share S_1 and p,h, γ_1', γ_2', and γ_3' should be embedded into share S_2, as shown in Figure 9.

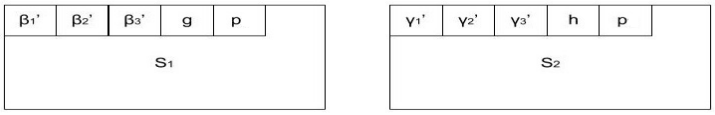


Figure 9 Structure of the shares.

4.2 Decoding Process

Since the decoding process should be done for authenticating the customer (participant P_2), the customer (participant P_2) extracts the modified 3OP parameter of share S_2 and calculates the original 3OP parameters $\gamma_1 = \gamma_1' + \alpha$, $\gamma_2 = \gamma_2' + \alpha$, and $\gamma_3' = \gamma_3 + \alpha$, where α should be agreed upon and kept secret by both parties. Finally, the ordinates of the 3OP are calculated using Eqs. (5) and (6). After obtaining the 3OP ordinates, the three points should be normalized as shown in Figure 10, where (x_1, y_1), (x_2, y_2), and (x_3, y_3), while (x_{1N}, y_{1N}) and (x_{3N}, y_{3N}) are the normalized two points of the 3OP.

Furthermore, the bank officer stacks share 2 and share 1 using the three normalized points. The normalizing process starts by finding the longest distance between the points of the 3OP and drawing a line between them. The point that is not on the line is set as the orthocenter of the 3OP (in Figure 10, the orthocenter is (x_2, y_2)). Furthermore, the two other points should be normalized by projecting the point onto the x axis and the y axis. In Figure 10, (x_1, y_1) is projected onto (x_{1N}, y_{1N}) and $(x_3 y_3)$ is projected onto (x_{3N}, y_{3N}). Finally, the two shares should be stacked by stacking the orthocenter, continuing with the other two points of the 3OP. Then it is checked whether the three points are aligned precisely. If these points are not aligned precisely, share 2 should be rotated until the right position is found and the secret image is revealed. When the secret image has appeared, the bank officer can authorize the customer.

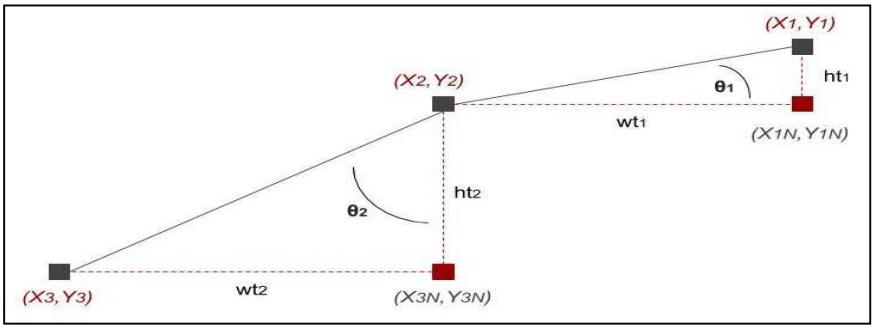


Figure 10 Normalization of 3OP.

4.3 Time Complexity

Based on the discussion in Section 4.1 (encoding process), it can be concluded that the time complexity for the encoding process of the proposed method is equal to $O(A\times(m+r))$ (where m is the pixel expansion, A is the area of the largest share, and r is the rotation interval). Meanwhile, the time complexity of Liu, *et al.*'s method is $O(m\times A)$. However, the time complexity for the decoding process of the proposed method is $O(m\times AI)$ (where m is the pixel expansion and AI is the area of the intersection of the two shares), while the time complexity of Liu, *et al.*'s method is $O((m\times A)^2)$ because their method uses brute force for the alignment process. Thus, the time complexity of the proposed method is smaller than that of Liu *et al.*'s method for the decoding process but the same for the encoding process. This is because the encoding process of the proposed method and Liu ,*et al.*'s one are similar.

5 Experiments and Discussion

This section discusses the result of the experiments that were conducted for evaluating the encoding and decoding time as well as the security analysis.

5.1 Experiments for Encoding and Decoding Process

Two experiments were conducted, one for comparing the performance of the encoding and decoding processes between the proposed method, Moni-Naor's method and Liu, *et al.*'s method. For evaluating the encoding process, share size was used as the independent variable while the encoding time was used as the dependent variable, using one secret image. The result is shown in Figure 11. It shows that the encoding time of the proposed method was longer than that of Liu, *et al.*'s and Moni-Naor's method because in the proposed method there is the requirement that the 3OP generated by the system should generate the intersection area. Based on the result of the experiment, the encoding time of the proposed method is in line with the encoding time complexity as discussed in Section 4.3, i.e. $O((m+r)\times A)$, while Liu, *et al.*'s and Moni-Naor's method has time complexity $O(m\times A)$.

For evaluating the decoding process, an experiment using the intersection area as the independent variable and the decoding time as the dependent variable was used. The experiment's result is shown in Figure 12. Based on Figure 12, it can be concluded that the decoding time of Moni-Naor's and Liu, *et al.*'s method are not influenced by the intersection area, while in the case of the proposed method the decoding time is influenced by the intersection area. This is because the stacking time depends on the intersection area instead of the largest share area. However, even if the decoding time of the proposed method depends on the intersection area, it is still shorter than the decoding time of Liu, *et al.*'s and

Moni-Naor's method. This is in line with the decoding time complexity for the proposed method as discussed in Section 4.3, i.e. $O(AI \times m))$, while Liu *et al.*'s and Moni-Naor's method have time complexity $O(m \times A)^2$.

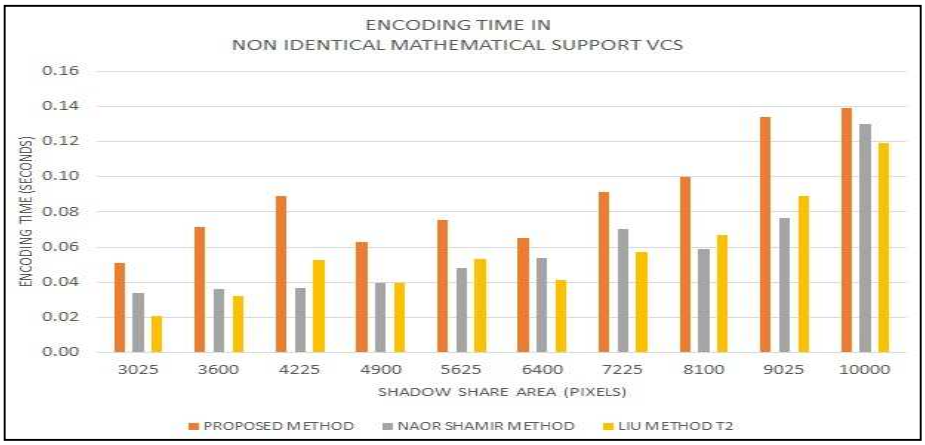


Figure 11 Encoding time for non-identical mathematical support VCS.

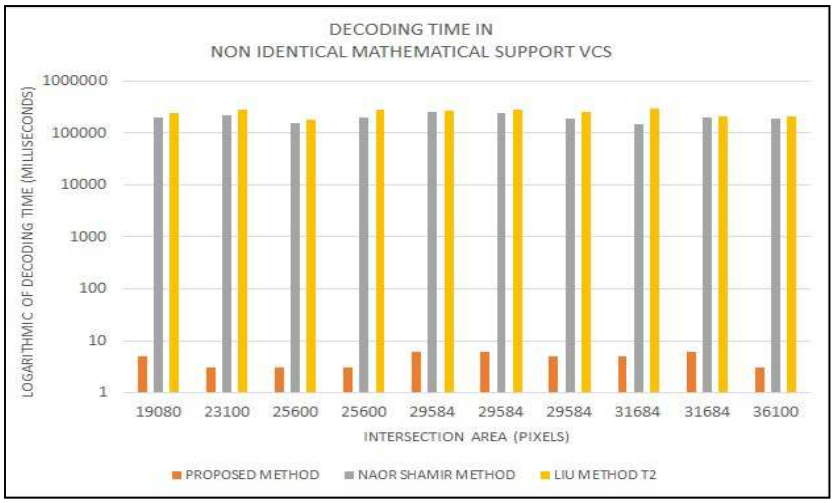


Figure 12 Decoding time for non-identical mathematical support VCS.

5.2 Security Analysis

Each share has different mathematical support, such that the share (C), the permutation code (K), and the secret image (M) are independent. Even if an attacker knows the share area as well as the permutation code, he/she cannot learn anything from those parameters. This is because the exact position of the

shares is unknown. In other words, the advantage of non-identical mathematical support VCS is that it not only secures the secret image but also the basis matrices structure.

The security analysis was done by analyzing the entropy [18,19]. Suppose the reduced 3OP parameters embedded into the shares is E, then it should be ensured that the parameters do not give any information about the messages by evaluating whether $H(M|E)$ is equal to 1 using the following equation:

$$H(M|E) = \frac{H(M)\times H(E)}{H(E)} = H(M) = 1 \quad (8)$$

Furthermore, $H(E|M) = H(E)$ should be proven for ensuring that the secret image does not give any information about the parameters using Eq. (9) as follow:

$$H(E|M) = \frac{H(E)\times H(M)}{H(M)} = H(E)=1 \quad (9)$$

Finally, it should be ensured that there is no mutual information between the parameters and the secret image using Eq. (10) as follow:

$$I(M;E) = H(M) - H(M|E) = 1 - 1 = 0 \quad (10)$$

Since $I(M;E) = 0$, it can be concluded that there is no mutual information between the reduced 3OP parameters and the secret image, such that if an attacker knows the parameters, he/she cannot obtain any information that is related to the secret image.

6 Conclusion

The main contribution of this study is the proposed method, called the Three-Orthogonal-Point (3OP) method. The three orthogonal points are used for reducing the decoding (alignment) time complexity of Naor Shamir's and Liu *et al.*'s methods. Using these points, the participants obtain the precise coordinates and orientation for each share such that the decoding process is speeded up. Based on the experimental result and discussion, it was proven that the proposed method is able to overcome the alignment problem in non-identical mathematical support visual cryptography schemes while maintaining security.

References

[1] Hegde, C., Manu, S., P Deepa Shenoy, P.D., Venugopal, K.R. & Patnaik, L.M., *Secure Authentication Using Image Processing and Visual Cryptography for Banking Applications,* in16th International Conference on Advanced Computing and Communications, pp. 65-72, 2008.

[2] Ching-Sheng, H. & Shu-Fen T. *Digital Watermarking Scheme with Visual Cryptography,* in International Multi Conference of Engineers and Computer Scientists, pp. 659-662, 2008.

[3] Liu, F., Wu, C.K. & Lin, X.J., *The Alignment Problem of Visual Cryptography Schemes*, Designs, Codes and Cryptography, **50**(2), pp. 215-227, 2009.

[4] Naor, M. & Shamir, A., *Visual Cryptography,* in EUROCRYPT'94, pp. 1-12, 1995.

[5] Chow, Y., Susilo, W. & Wong, D.S., *Enhancing the Perceived Visual Quality of a Size Invariant Visual Cryptography Scheme,* in Information and Communications Security, pp. 10-21, 2012.

[6] Naor, M. & Pinkas, P., *Visual Authentication and Identification,* in CRYPTO'97, pp. 322-336, 1997.

[7] Stinson, D., *Visual Cryptography and Threshold Schemes*, IEEE Potentials, **18**(1), pp. 13-16, 1999.

[8] Machizaud, J., Chavel, P. & Fournel, T., *Fourier-based Automatic Alignment for Improved Visual Cryptography Schemes*, Optics Express, **19**(23), pp. 14-15, 2011.

[9] Liu, F. & Yan, W.Q., *Visual Cryptography for Image Processing and Security*, Springer International Publishing, pp. 23-61., 2014.

[10] Weir, J. & Yan, W. *Resolution Variant Visual Cryptography for Street View of Google Maps*, in IEEE International Symposium on Circuits and Systems, pp. 1695-1698, 2010.

[11] Yan, W., Duo J. & Kankanhalli, M.S., *Visual Cryptography for Print and Scan Applications*, in Proceedings of IEEE International Symposium on Circuits and Systems, pp. 572-575, 2004.

[12] Yang, C.N., Chen, T.S., *Size-adjustable Visual Secret Sharing,* Science, **E88-A**(9), pp. 2471-2474, 2005.

[13] Ateniese, G., Blundo, C., Santis, A.D. & Stinson, D.R., *Extended Capabilities for Visual Cryptography*. ACM Theory of Computer Science, **250**(1-2), pp. 143-161, 2001.

[14] Jin, D., *Progressive Color Visual Cryptography,* School of Computing, Masters Thesis, NUS, 2003.

[15] Chen, Y.F., Chan, Y.K., Huang, C.C., Tsai, M.H., Chu, Y.P., *A Multiple-level Visual Secret-sharing Scheme Without Image Size Expansion,* Information Sciences **177**, pp. 4696-4710, 2007.

[16] Krawczyk, H. & Rabin, T., *Chameleon Hashing and Signatures*, IACR Cryptology ePrint Archive, retrieved on 12 Janaury 2017, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.3262&rep=rep1&type=pdf,1998.

[17] Floyd, R. & Steinberg, L., *An Adaptive Algorithm for Special Grayscale,* in Proc. of the Soc. for Information Display, pp. 75-77, 1976.

[18] Dumas, J., Roch, J., Tannier, É. & Varrette, S., *Foundations of Coding: Compression, Encryption, Error Correction*, John Wiley & Sons, 2015.
[19] Shannon, C.E., *A Mathematical Theory of Communication,* Mobile Computing and Communications Review, **5**(1), pp. 3-55, 2001.