PYTHON IMPLEMENTATION OF VISUAL SECRET SHARING SCHEMES

Ruxandra Olimid
Faculty of Mathematics and Computer Science,
University of Bucharest
Email: **ruxandra.olimid@fmi.unibuc.ro**

Abstract

Visual secret sharing schemes (VSS) represent an important concept of visual cryptography. They permit the sharing of a secret image between multiple participants so that only authorized groups can recover the secret.

This paper considers the software implementation of some black-and-white secret images VSS in Python programming language. PIL (Python Imaging Library) provides strong image processing capabilities, making the library suitable for this kind of implementation. We present samples of the results obtained from the software computation and draw some conclusions.

Keywords: visual secret sharing, visual cryptography, Python, PIL (Python Imaging Library)

1. Introduction

A secret sharing scheme permits the sharing of a secret between multiple participants so that only authorized groups can recover the secret. Even by putting their shares together, the members of an unauthorized group are not able to reveal the secret or significant information about it.

Visual secret sharing schemes (VSS) represent the particular case for which the secret and therefore the shares are images. Naor and Shamir were the first to introduce them as a part of visual cryptography [NAOR94]. Since then, multiple models of VSS were defined, for black-and-white, grayscale or color pictures. Examples from the literature includes [ATEN96] and [ITO99].

This paper exemplifies and analyses the implementation of some of the existing VSS using Python's Image Library (PIL). Section 2 defines the VSS that were considered for the implementation. Section 3 introduces the resources and techniques used during development. Section 4 exposes computational samples and data analysis. Finally, we conclude.

2. Visual Secret Sharing Schemes

This section introduces the VSS that were considered for the software implementation. We restrict our interest to black-and-white secret images, but a similar approach can be considered for color pictures as well.

In most cases, the type of the shares copies the type of the secret, in the sense that for a black-and-white secret, the shares are also black-and-white images. However, Ito and al. introduced a way to share a black-and-white secret image into colored images components [ITO99]. A similar, but simpler secret scheme that follows the same idea will be considered [OLIM10].

Each image (secret or share) is considered to be a matrix of pixels. By convention, in black-and-white representation, a white pixel is represented by 0 and a black pixel is represented by 1. For color shares, only R (red), G (green) and B (blue) pixels are used (no other combination of RGB colors can appear in a share).

For the rest of the paper, $P = \{P_1,..., P_n\}$ denotes the set of participants and therefore n denotes their number.

Definition 1. The collection of all authorized sets of participants to reconstruct the secret is called the *access structure (A)*. The collection of all forbidden sets of participants to access the secret is denoted by NA.

Definition 2. The *collection of the maximal forbidden sets* $NA_{\max}$ is defined as

$$NA_{\max} = \{B \in NA \mid \forall B' \in NA \setminus B, B \not\subset B'\}$$

Definition 3. A secret sharing scheme is *unanimous* (or *(n,n) secret sharing scheme*) if all n shares are needed in order to reconstruct the secret.

In case of a unanimous secret sharing scheme, the access structure contains only 1 set, the set of all participants. This represents a particular type of access structure. If no restrictions are required, then the access structure is called *general access structure*.

The paper considers 3 types of black-and-white secret image VSS:

1) *unanimous visual secret sharing scheme;*
2) *general access structure secret sharing scheme;*
3) *color shares VSS.*

2.1. Naor-Shamir Unanimous VSS

Naor and Shamir were the first to introduce a visual secret sharing scheme [NAOR94].

We will restrict to their unanimous scheme, the particular case in which all participants must agree to reconstruction. If at least one participant does not agree to share its component, the secret image is perfectly secured (the others find no information about the secret).

1) *Computing shares*

   Consider $W = \{e_1, e_2,..., e_n\}$ where n is the number of participants and e_i is the n-element vector with 1 on position i and 0 otherwise.

   Let:

   - $\pi_1, \pi_2,..., \pi_{2^n-1}$ be the even cardinality subsets of W;
   - $\sigma_1, \sigma_2,..., \sigma_{2^n-1}$ be the odd cardinality subsets of W.

   Each set defines a $n \times 2^{n-1}$ matrix, $S^0 = (S^0_{ij})$ and respectively $S^1 = (S^1_{ij})$:

   - $S^0_{ij} = 1 \Leftrightarrow e_i \in \pi_j, i = 1..n, j = 1..2^{n-1}$;
   - $S^1_{ij} = 1 \Leftrightarrow e_i \in \sigma_j, i = 1..n, j = 1..2^{n-1}$;

   Consider:

   - C_0, the set of matrices obtained by permuting the columns of S^0;
   - C_1, the set of matrices obtained by permuting the columns of S^1.

To each pixel in the secret image will correspond 2^{n-1} pixels in each share:

- if the pixel is white, an element from C_0 is randomly chosen. The corresponding pixels in $share_i$ are given by row_i of the selected matrix;
- if the pixel is black, an element from C_1 is randomly chosen. The corresponding pixels in $share_i$ are given by row_i of the selected matrix.

2) *Reconstruction of the secret image*
All shares are added pixel by pixel (or the corresponding bits are OR -ed).

Theorem 1. The above scheme is a unanimous scheme with n participants, where $k = 2^{n-1}$ is the number of pixels in each share that correspond to a pixel in the secret; $\alpha = 1/2^{n-1}$ is the contrast parameter; $r = 2^{n-1}!$ is the cardinal of C_0 and C_1 [NAOR94].

In order to obtain the exact original secret image, additional transformations are needed. The 2^{n-1} pixel groups should be transformed in a black pixel, if the number of 1s is greater than the contrast parameter, or white, otherwise. For the rest of the paper, we will not consider this improvement.

Example 1. *Naor-Shamir unanimous VSS for* $n = 2$ *participants.*

From the construction algorithm, the values of S^0 and S^1 are:

$$S^0 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}; S^1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The possible corresponding shares of a pixel are summarized in Table 1.

White pixel □	First share	□■	■□
	Second share	□■	■□
	Result	□■	■□
Black pixel ■	First share	■□	□■
	Second share	□■	■□
	Result	■■	■■

Table 1. All possible shares in Naor-Shamir unanimous VSS with 2 participants

Example 2: *Naor-Shamir unanimous VSS for* $n = 3$ *participants.*

From the construction algorithm, the values of S^0 and S^1 are:

$$S^0 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}; S^1 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Some of the possible corresponding shares of a pixel are displayed in Table 2.

White pixel □	First share	□■■□	…	□■■□
	Second share	□■□■	…	■□■□
	Third share	□□■■	…	■■□□
	Result	□■■■	…	■■■□
Black pixel ■	First share	■□□■	…	■□□■
	Second share	□■□■	…	■□■□
	Third share	□□■■	…	■■□□
	Result	■■■■	…	■■■■

Table 2. Some possible shares in Naor-Shamir unanimous VSS with 3 participants

White pixel □	First share	□□ ■■	■■ □□	□■ □■	■□ ■□	□■ ■□	■□ □■
	Second share	□□ ■■	■■ □□	□■ □■	■□ ■□	□■ ■□	■□ □■
	Result	□□ ■■	■■ □□	□■ □■	■□ ■□	□■ ■□	■□ □■
Black pixel ■	First share	□□ ■■	■■ □□	□■ □■	■□ ■□	□■ ■□	■□ □■
	Second share	■■ □□	□□ ■■	■□ ■□	□■ □■	■□ □■	□■ ■□
	Result	■■ ■■	■■ ■■	■■ ■■	■■ ■■	■■ ■■	■■ ■■

Table 3. All possible shares in Naor-Shamir unanimous VSS with 2 participants that maintains image ratio constant

Example 3: *Naor-Shamir unanimous VSS for* $n = 2$ *participants that maintains the ratio of the secret image constant*

In previous examples, in order to represent a pixel, one dimension increases k times. To avoid the distortion of the image, both dimensions can be increased by the same amount. Nevertheless, the dimension of a share becomes larger (Table 3).

2.2. General VSS

The previous schemes present the special property of unanimity. Ateniese and al. introduced a model of VSS that permits general access structures [ATEN96].

Definition 4. Be $P = \{P_1, ..., P_n\}$ the set of all participants and $NA_{\max} = \{B_1, ..., B_r\}$ the collection of the maximal forbidden sets. The *cumulative array* of the access structure A is defined as a $n \times r$ matrix $C_A = (b_{ij})_{1 \le i \le n, 1 \le j \le r}$, where:

$$b_{ij} = \begin{cases} 0, P_i \in B_j \\ 1, P_i \notin B_j \end{cases}$$

The cumulative array specifies if a participant P_i belongs to an unauthorized maximal set B_j, by setting the appropriate element to 0.

A scheme that allows a general access structure can be defined staring from the Naor-Shamir (n, n) VSS:

1) *Computing shares*

Let $P = \{P_1, ..., P_n\}$ be the set of participants, A the access structure, $NA_{\max}$ the collection of maximal forbidden sets, t the cardinal of $NA_{\max}$ and C_A the cumulative array. S^0 and S^1 are defined as the matrixes from the unanimous Naor-Shamir scheme for t participants.

For each fixed i $(1 \le i \le n)$ consider $J_i = \{ j \mid C_A(i, j) = 1\}$ and define:

- $\overline{S}^0$ as the matrix with row_i equal to the result of applying OR to all $row_j, j \in J_i$ of matrix S^0;
- $\overline{S}^1$ as the matrix with row_i equal to the result of applying OR to all $row_j, j \in J_i$ of matrix S^1.

Consider:

- C_0 , the set of matrices obtained by permuting the columns of $\overline{S}^0$;
- C_1 , the set of matrices obtained by permuting the columns of $\overline{S}^1$.

To each pixel in the secret image will correspond 2^{n-1} pixels in each share:

- if the pixel is white, an element from C_0 is randomly chosen. The corresponding pixels in $share_i$ are given by row_i of the selected matrix;
- if the pixel is black, an element from C_1 is randomly chosen. The corresponding pixels in $share_i$ are given by row_i of the selected matrix.

2) *Reconstruction of the secret image*

All shares are added pixel by pixel (or the corresponding bits are OR -ed).

Example 4. *A general VSS for* $n = 4$ *participant, and the collection of the maximal forbidden sets* $NA_{\max} = \{\{P_1, P_2\}, \{P_2, P_3\}, \{P_3, P_4\}\}$.

From construction:

$$C_A = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

As the cardinal of $NA_{\max}$ is 3, then the corresponding matrixes $\overline{S}^0$ and $\overline{S}^1$ are build starting from S^0, respectively S^1 defined in Example 2:

$$\overline{S}^0 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}; \overline{S}^1 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

An example of sharing a white, respectively a black pixel is shown in Table 4.

White pixel	First share	Second share	Third share	Forth share
□	□■■■	□□■■	□■■□	□■■■
Black pixel	First share	Second share	Third share	Forth share
■	□■■■	□□■■	■□□■	■■□■

Table 4. A possible sharing for a white and a black pixel for the VSS in Example 4

2.3. Black-and-white secret image VSS with RGB shares

Ito, Kuwakado and Tanaka used color components in order to share a black-and-white secret image [ITO99]. We will consider here a scheme that uses the same idea, but the pixels color (red, green or blue) in the components is (almost) randomly chosen from R, G and B [OLIM10]. The usage of randomness eliminates the need of the fixed matrices, idea inherited from Naor-Shamir VSS.

The scheme is available in the additive RGB model, in which white results by adding red, green and blue together at maximum intensity, under the assumption that adding any other color to white it remains white. Black is considered to be the absence of any color.

1) Computing shares

The pixels of the secret shares are randomly choose from {R, G, B} so that:

- if the pixel of the secret image is white, then there must exist a component with the corresponding pixel R, another component with the corresponding pixel G and a third component, different from the previous two, with the corresponding pixel B;

- if the pixel of the secret image is black, then all the corresponding pixels of the shares are randomly choose from {R, G}, from {R, B} or from {G, B} so that there exist at least 2 pixels of different colors.

2) *Reconstruction of the secret image*
   - *k* users try to reconstruct the secret by overlapping their correspondent shares (the shares are added together, pixel by pixel);
   - each colored computed pixel in the resulting image is transformed into a black pixel. White pixels remain unchanged.

4 examples of sharing are given for a white and respectively a black pixel in Table 5. Please notice that for each white pixel exist at least a red share, a green share and a blue share, while for a black pixel, the corresponding shares have only 2 colors (R&G or R&B or B&G).

4 white pixels	First share	Second share	Third share	Forth share	Fifth share	Reconstructed color image
⬜⬜ ⬜⬜	RR RR	GR RB	BB BG	RG GG	RB GB	WW WW
4 black pixels	First share	Second share	Third share	Forth share	Fifth share	Reconstructed color image
⬛	RR BB	GB BB	RR GB	RR GG	RR GG	YM CC

Table 5. A possible sharing for 4 white and 4 black pixels in a scheme with 5 participants
(R = Red, G = Green, B=Blue, Y = Yellow, M = Magenta, C = Cyan, W=White)

3. Development environment, implementation methods and techniques

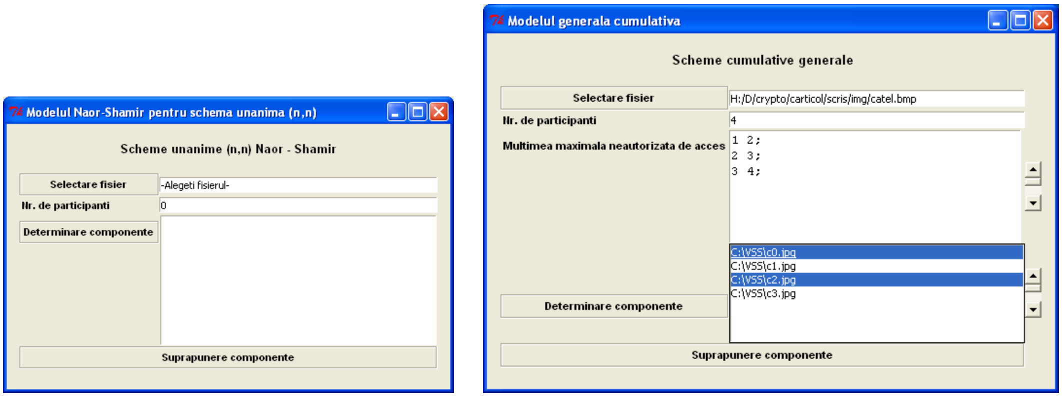
The previously mentioned VSS algorithms were implemented in Python programming language using IDLE (Python's Integrated DeveLopment Environment 2.6.6) as the development environment [PYTH11].

Python is a powerful dynamic programming language, available for all major operating systems and available under an open source license. Besides other benefits, Python was mainly chosen due to the fact that it provides a powerful and easy to use image library, named PIL (Python Image Library), also available in a free version.

PIL provides image processing capabilities, available for different kind of file formats. A fully documentation can be found at [PIL11].

Some of the special features that PIL provides and were used during the implementation include:

- Different image *modes*, defining the type and depth of a pixel in the image. Even though it supports multiple standard modes, for the current implementations were used:
  - *1* (1-bit pixels, black and white, stored as 8-bit pixels);
  - *RGB* (3x8-bit pixels, true color).
- The possibility to consider the image as a matrix of pixels, permitting the modification at pixel level, setting the pixel value, etc.;


Modelul Naor-Shamir pentru schema unanima (n,n)
Scheme unanime (n,n) Naor - Shamir
Selectare fisier
Determinare componente
Suprapunere componente
Modelul generala cumulativa
Scheme cumulative generale
Selectare fisier
Determinare componente
Suprapunere componente

Selectare fisier: -Alegeti fisierul-

Nr. de participanti: 0

Selectare fisier: H:/D/crypto/carticol/scris/img/catel.bmp

Nr. de participanti: 4

Multimea maximala neautorizata de acces:

```
1 2;
2 3;
3 4;
```

```
C:\VSS\c0.jpg
C:\VSS\c1.jpg
C:\VSS\c2.jpg
C:\VSS\c3.jpg
```

Figure 1. Graphical interfaces for 2 implemented VSS

- Defining or reading the *size* of an image as a 2-tuple consisting in the horizontal and vertical size in pixels (*<image>.size*);
- Working with bands of colors (R, G and B bands for color images);
- Using image functions as: opening an existing image (*image.open(<image>, <mode>)*), creating a new image (*image.new(<mode>,<size>)*), creating an image based on multiple bands of color (*image.merge(<mode>, (<band1>, <band2>, <band3>))*).

Python also provides other functions that were mandatory for the implementation of the VSS as:

- intertools module, that standardizes efficient and useful tools for combinatoric generators as permutations (*intertools.permutations(<p>)*) or combinations (*itertools.combinations(<n>,<k>)*);
- random module, that implements pseudo-random generators: random.randrange([<start>], <stop>, <step>) returns a randomly selected element in the specified range.

All the previously mentioned schemes were implemented within graphical interfaces (Figure 1) that permit:

- to receive the secret image as the input file;
- to fill in all the necessary parameters, depending on the selected VSS (for example the number of participants for Naor-Shamir unanimous VSS or the number of participants and the collection of maximal forbidden sets for general access structure VSS);
- to compute the shares accordingly to the filled in inputs;
- to reconstruct the image based on the selected shares (the selection of different set of shares leads to different reconstructed images, depending on the authorization of the group).

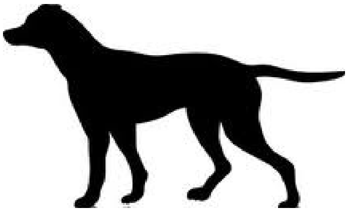


Figure 2. Input test image

4. Software computation and data analysis

The image in Figure 2 was given as the secret image input for all the implemented VSS models. The other inputs are specified below:

- Naor-Shamir unanimous VSS was tested for $n = 3$ participants (Figure 3);
- Naor-Shamir unanimous VSS that maintains image ratio constant was tested for $n = 2$ participants (Figure 4);
- The cumulative VSS was tested as in Example 4, with $n = 4$ participants and $NA_{\max} = \{\{P_1, P_2\}, \{P_2, P_3\}, \{P_3, P_4\}\}$ (Figure 5);
- Black-and-white secret image VSS with RGB shares was tested for $n = 4$ participants (Figure 6).

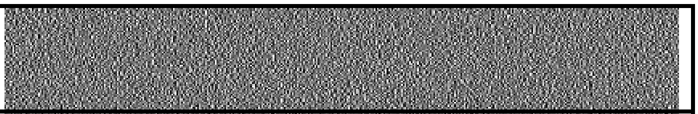
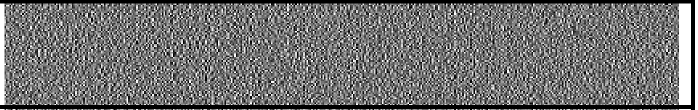
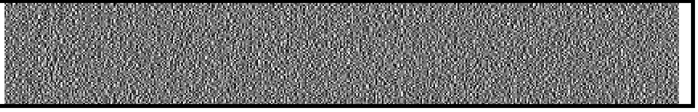
Share 1	
Share 2	
Share 3	

Figure 3.a. The shares in the Naor-Shamir unanimous VSS (n = 3)

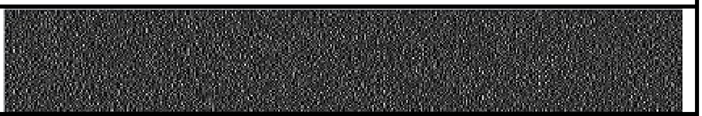
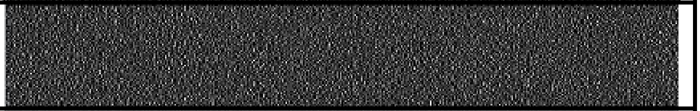
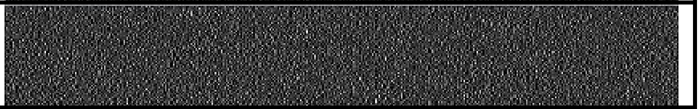
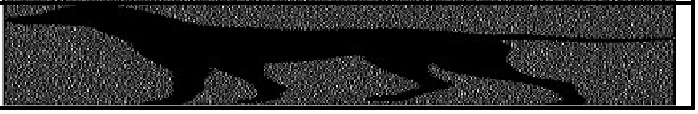
Combined shares	Computed result
Share 1 Share 2	
Share 1 Share 3	
Share 2 Share 3	
Share 1 Share 2 Share 3	

Figure 3.b. Computed images by using all possible sets of shares in the Naor-Shamir unanimous VSS (n = 3)

It is easy to remark that the practical implementation supports the theory results. In case of Naor-Shamir unanimous VSS:

- The secret image is reconstructed only by combining together all the shares. No information is leaked when fewer participants cooperate;
- The reconstructed image increases the length of the original image;
- The contrast of the reconstructed image is decreasing (white becomes gray);

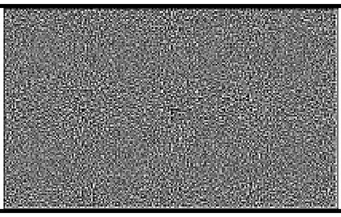
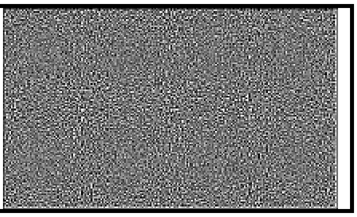
Share 1		Share 2	

Figure 4.a. The shares in Naor-Shamir unanimous VSS ($n = 2$) that maintains the image ratio

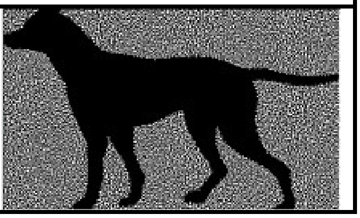
Combined shares	Computed result
Share 1 Share 2	

Figure 4.b. The reconstructed image from the shares in the Naor-Shamir unanimous VSS ($n = 2$) that maintains the image ratio constant

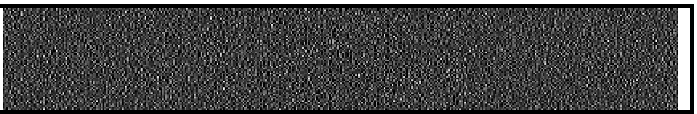
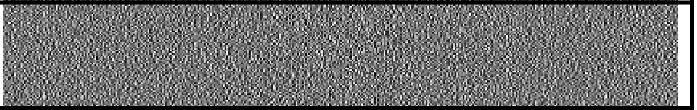
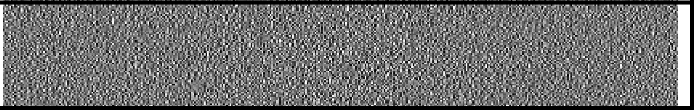
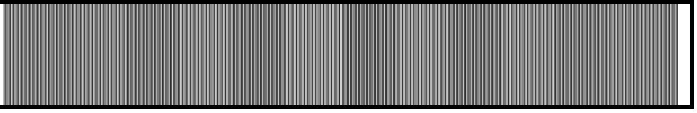
Share 1	
Share 2	
Share 3	
Share 4	

Figure 5.a. The shares in the cumulative VSS ($n = 4$)

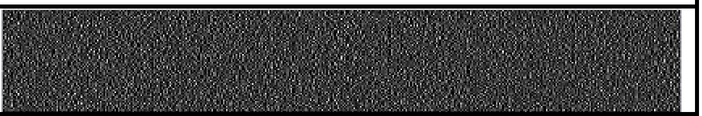
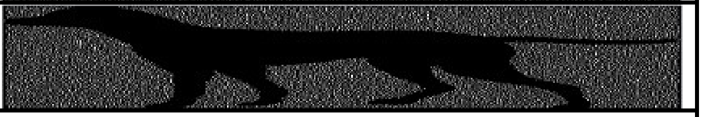
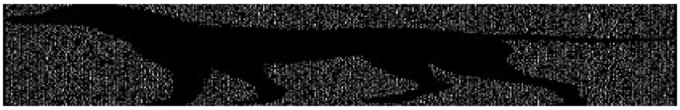
Combined shares	Computed result
Share 1 Share 2	
Share 1 Share 2 Share 3	
Share 1 Share 2 Share 3 Share 4	

Figure 5.b. Sample of computed images by using different sets of shares in the cumulative VSS ($n = 4$)

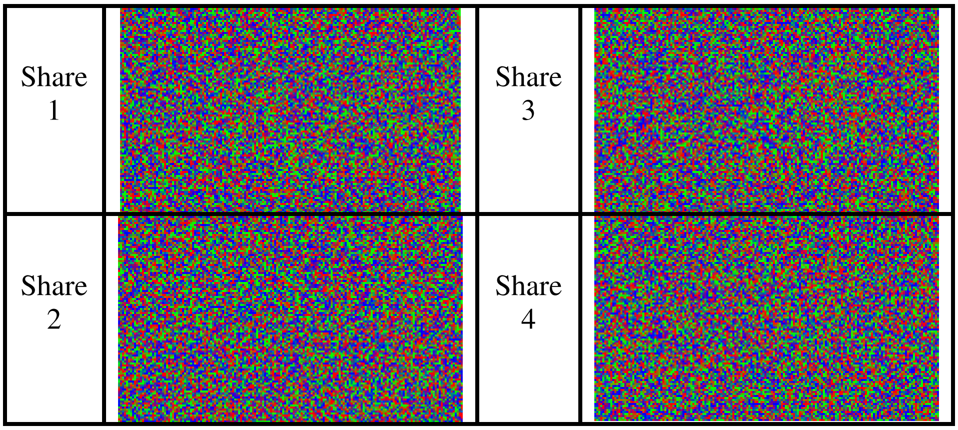


Figure 6.a. Shares in black-and-white secret image VSS with RGB shares ($n = 4$)

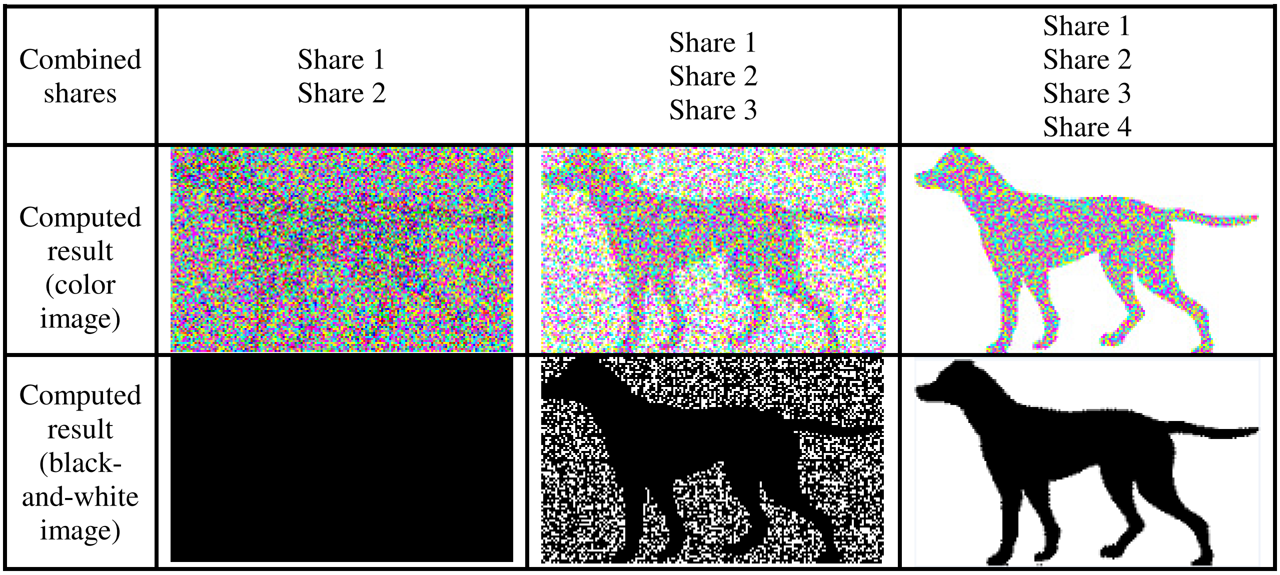

Combined shares
Share 1 Share 2
Share 1 Share 2 Share 3
Share 1 Share 2 Share 3 Share 4
Computed result (color image)
Computed result (black-and-white image)

Figure 6.b. Sample of computed images by using different sets of shares in black-and-white secret image VSS with RGB shares ($n = 4$)

In case of Naor-Shamir unanimous VSS that maintains the image ratio constant:

- The image ratio is maintained constant;
- All the other properties of the Naor-Shamir unanimous VSS are preserved: contrast is smaller than in the original image and the scheme is perfect in the sense that it provides no information about the secret when the number of cooperating participants is less than *n*.

In the case of the cumulative general access structure VSS:

- The contrast of the reconstructed image is decreasing;
- $\{P_1, P_2\}$ is a maximal forbidden group, so they are not able to reconstruct the secret image;

- By joining any other participant to a maximal forbidden group, they become able to reconstruct the secret: for example $\{P_1, P_2, P_3\}$ can recover the secret image (Figure 5.b);
- Because $\{P_1, P_2, P_3, P_4\}$ extends the maximal forbidden set $\{P_1, P_2\}$ (or $\{P_2, P_3\}$, or $\{P_3, P_4\}$), it also represents an authorized group of participants. But even though the number of shares is larger than in case of the set $\{P_1, P_2, P_3\}$, the reconstructed image is less similar than the original.

In the case of the black-and-white secret image VSS with RGB shares:

- The dimension of the shares is equal to the dimension of the secret, but more information is needed for storage because of the color information;
- The reconstructed image maintains the ratio of the initial secret image;
- Any 2 shares provide no information about the secret image (the reconstructed image is totally black);
- More than 2 shares provides enough information (all black pixels are correctly reconstructed);
- All shares lead to the perfect reconstruction of the initial secret image.

5. Conclusions

The paper presents a possible implementation of some visual secret schemes (VSS) under the Python programming language, using the PIL library. Data inputs, computational results and a succinct analysis are performed for each considered scheme.

References:

[ATEN96] Giuseppe Ateniese, Carlo Blundo, Alfred De Santis, Douglas R. Stinson,"Visual Cryptography for General Access Structures", Electronic Colloquium on Computational Complexity (ECCC), vol.3, no.12, 1996.

[GHOD98] Hossein Ghodosi, Josef Pieprzyk, Rei Safari-Navini, Huaxiong Wang, "On construction of Cumulative Secret Sharing Schemes", LNCS, 1998, pg.379-390.

[ITO99] Ryo Ito, Hidenori Kuwakado, Hatsukazu Tanaka, "Image Size Invariant Visual Cryptography", IEICE Trans. On Funda. Of Elect., Comm. And Comp.Sci., Vol.E82-A, No.10(1999), pg.2172-2177.

[NAOR94] Moni Naor, Adi Shamir, "Visual Cryptography", Advanced in Cryptology – EUROCRYPT'94, LNCS950, Springer-Verlag(1995), pg.1-12.

[OLIM10] Ruxandra Olimid, "About a Visual Secret Sharing Scheme", Proceedings of the 3rd International Conference on Security for Information Tehnology and Communications, Bucharest, 2010, pg. 13-17.

[PIL11] Python Image Library (PIL) http://www.pythonware.com/products/pil/

[PYTH11] Python Programming Language – Official Website http://www.python.org/