An efficient lossless (2, n) secret image sharing based on Blakley's scheme

Sebastien Beugnon, William Puech, Jean-Pierre Pedeboy

An efficient lossless (2, n) Secret Image Sharing based on Blakley's Scheme

Sébastien BEUGNON*† William PUECH* Jean-Pierre PEDEBOY †

*LIRMM, UMR 5506 - Université de Montpellier
860 rue de St-Priest, 34095 Montpellier, FRANCE
Email: {sebastien.beugnon,william.puech}@lirmm.fr

Stratégies S.A.
56 Rue d'Arcueil, 94578 Rungis, FRANCE

***Abstract*—Visual Secret Sharing (VSS) is a type of cryptographic method used to secure digital media such as images by splitting it into n shares. Then, with k or more shares, the secret media can be reconstructed. Without the required number of shares, they are totally useless individually. The purpose of secret sharing methods is to reinforce the cryptographic approach from different points of failure as a single information-container. Instead of Lagrange's interpolation employed by classical methods, an approach based on linear equations and coding theory resolves the classic problems of VSS methods like pixel expansion and information losses at recovery time. In this paper, we propose an efficient (2, n) secret image sharing scheme without loss of information and size increase of shares, with $n \leq 7$ based on linear equations encoded into eight bits to encode separately pixels and permutations to maintain high entropy in shares.**

I. Introduction

With the democratization and the growth of the Internet, web, social networks and other multimedia technologies, a huge amount of digitized information such as images, audio and videos are produced everyday. The need to secure sensitive data from unauthorized access and protect information from piracy, is becoming a major topic in this digital age, specifically for sensitive domains such as; military, medical and governmental. To prevent vulnerabilities of information systems and networks, many methods were proposed like image data hiding and watermarking. The major drawbacks of these methods [1] are their computational time and the weakness introduced by the use of single information container. Secrets can be lost or even worse tampered with, making it impossible to read the encrypted data. Duplicated containers only increase the danger of security exposure.

Secret sharing's concept was introduced in 1979 by Blakley [2] and Shamir [3] independently to solve this dilemma. Both methods based themselves on a (k, n) threshold scheme, where n is the number of generated shares and only groups of k or more shares can recover the original secret. However, both methods focus mainly on binary data and do not take into consideration the content of the data.

Secret image sharing schemes are divided into two categories : Visual Cryptography and Computational Secret Sharing. The first visual cryptographic scheme (VCS) for images was proposed in 1994 by Naor and Shamir for binary images [4]. Each pixel of the image is encoded into n matrices of subpixels and to recover the secret, the decoding process combines matrices to show the secret image. However, VCS depend solely on human visual system and the first ones increase the size of shares in function of k. Whereas computational secret image sharing schemes need some computations to reconstruct the secret image, but allow new properties to emerge. Thien and Lin proposed in 2002 a very efficient scheme based on Shamir's scheme producing shares reduced to $1/k$ of the original size and resembling random noise thanks to a secret-key-controlled permutation [5]. Following work has concentrated more on reducing the size of shares or hiding them into meaningful images to enhance the confidentiality [6, 7, 8]. Blakley's scheme has also been used to share secret image, Tso proposes to quantize the pixel values to reduce drastically the size of shares and the quality of the reconstructed image [9]. Tsai and Chen present along their method some implementation of Blakley's scheme using a prime finite field [10]. Meanwhile, Chen *et al.* propose another approach based on the works of Blakley and Thien by restraining the size in bits of coefficients [11].

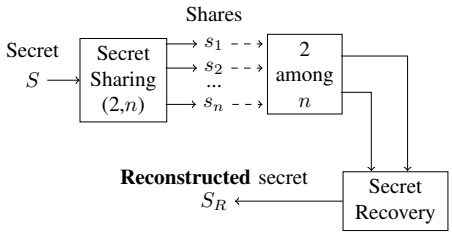

Figure 1. An example of (2, n) secret sharing scheme.

In this paper we propose a new (2, n) secret image sharing method to protect content as shown in Fig. 1. Our method, which does not require a secret key, is based on Blakley's scheme with a specific encoding, allowing to recover the secret image without loss and increasing the size of the shares. Moreover, each pixel of the secret image is encoded separately in each share. In Section II we provide a review of the Blakley's scheme. Then, in Section III, we develop the proposed method. Experimental results are presented in Section IV. Finally, in Section V we draw some conclusions.

978-1-5090-3649-3/17/$31.00 ©2017 IEEE

II. Review of Blakley's Scheme

In 1967, Blakley proposed a geometric approach to solve the problem of secret sharing [2]. His scheme defines the secret as a point in a k-dimensional space and shares as hyperplanes where the point embedding the secret must lie:

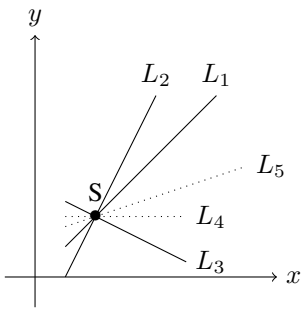

Figure 2. Example of a two dimensional Blakley's secret sharing scheme.

The secret is recovered by intersecting at least k hyperplanes as shown in Fig. 2 for a (2, n) Blakley's scheme. However, this method has some flaws. It is less space efficient than Shamir's scheme: the greater the k is, the larger the shares. To prevent this expansion, the scheme can be restrained on the number of usable hyperplanes for shares. By example, for a (2, n), we can use as definition of hyperplane of dimension 2, i.e. a line, the following equation:

$$y = A \times x + B, \text{ where } x \text{ is the secret.} \tag{1}$$

III. Proposed scheme

The proposed method is a $(2, n)$ secret image sharing scheme, where $2 \leq n \leq 7$, consisting of 3 steps which are the diffusion, the share generation and the permutation stream shuffle. This method does not require any supplementary key. This section describes the scheme process as illustrated in Fig. 3.

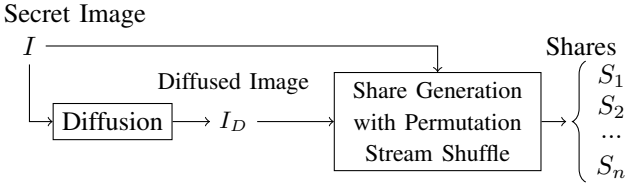

Figure 3. Overview of the method.

A. Diffusion phase

This step consists to spread the information across the secret image as a preprocessing transformation to increase the robustness of our scheme to histogram analysis and attacks using uniform image. The diffusion phase takes the form of a reversible gradient transformation:

$$p_{I_D}(i) = (p(i) + i) \bmod 256, \tag{2}$$

where $p(i)$ is the pixel value of the current pixel at the position i of the secret image array and $p_{I_D}(i)$ the pixel value of the diffused image at the same position.

B. Share generation

The sharing phase of our method consists in determining equations for each pixel of the diffused image p_{I_D}. Before that, for each possible pixel value in an image $\{0, ..., 255\}$, a pair of coordinates (x, y), where $x, y \in \{-8, ..., 7\}[\![n\ ;\ p]\!]$, is assigned:

$$\begin{cases} x = (p_{I_D}(i) \bmod 16) - 8, \\ y = \lfloor \frac{p_{I_D}(i)}{16} \rfloor - 8 \end{cases} . \tag{3}$$

These coordinates are used as intersection points of lines in our method.

Each equation of our scheme is encoded on only one byte in order to share independently each pixel of the secret image. These equations depend of 3 variables: a coefficient a, an offset l corresponding to the translation on x-axis applied to the equation and a coefficient b. The parameters a and l are encoded as a pair on 4 bits and b on 4 bits also as shown in Fig. 4.

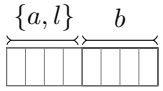

Figure 4. Encoded equation for a pixel share (one byte).

To uniformly represent the same number of equations for all the coordinates in $\{-8, ..., 7\} \times \{-8, ..., 7\}$, we use a specific set of equations defined by the following equation:

$$y = a \times x + -\text{sgn}(a) \times b - a \times (b + l), \tag{4}$$

where the function $\text{sgn}(a)$ is defined by the system:

$$\forall x \in \mathbb{R},\ \text{sgn}(x) = \begin{cases} -1 & \text{if } x \leqslant 0 \\ 1 & \text{otherwise} \end{cases} . \tag{5}$$

With the example illustrated in Fig. 5, we share a pixel as $p_{i_D}(i) = 189$, this pixel value is then converted into coordinates (5,3). Blue lines (L_0, L_1, L_2, L_3) represent examples of lines passing through the coordinates (5,3) which means equations of these lines encode the pixel value 189. While red lines represent two axes, respectively named: Axis_1 and Axis_0 ($y = x$ and $y = -x$). Instead of using the classic axes x and y, we use as foundation these two axes to simplify the representation of line equations in our scheme. Each equation defined in our method has to have an intersection point with one of the two axes.

Always with the same example, the line noted L_1 intersects Axis_0 at the point (1,-1). This intersection point noted $P =(x_p, y_p)$ has the following property: for all equations defining

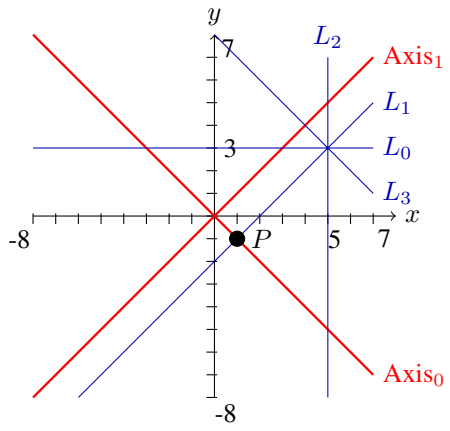

Figure 5. Example of line equations (L_0, L_1, L_2, L_3) used to share the point (5,3) and P the point of intersection of L_1 with Axis$_0$.

straight lines intersecting Axis$_1$ $\exists x \in \{-8, ..., 7\}, P = (x, x)$; while for Axis$_0$, $\exists x \in \{-8, ..., 7\}, P = (x, -x)$.

$$\begin{cases} y = a \times x + B \\ y_p = a \times x_p + B \end{cases} \Leftrightarrow \begin{cases} y = a \times x + (-a \times x_p + y_p) \\ B = y_p - a \times x_p \end{cases}. \tag{6}$$

With this construction, y_p depends only of the sign of coefficient a and of the value of x_p. So it can be reduced to $y_p = -\text{sgn}(a) \times x_p$. This way, we need to save the value of x_p as b along with the coefficient a.

Except the special case 0* that encodes a vertical line equation ($y = b$), all line equations use as coefficient a values from the set $\mathbb{C}_f = \mathbb{Z} \cup \{\forall i \in \mathbb{Z}^*, \frac{1}{i}\}$. However, it is not sufficient to represent all values in $\{-8, ..., 7\} \times \{-8, ..., 7\}$.

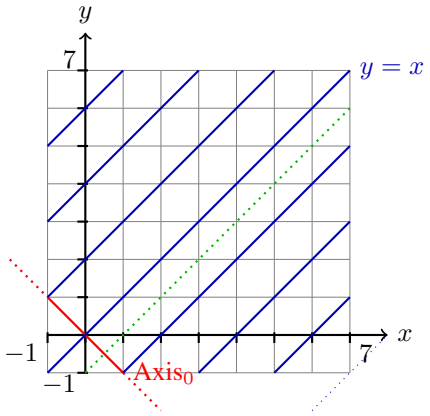

Figure 6. Example of line equations defined in our scheme: blue lines have coefficient $a = 1$. The green dotted line is the translation of blue line on x-axis of equation $y = x$ with $l = 1$.

As shown in Fig. 6, when $a = 1$ and $\forall x_p \in \{-8, ..., 7\}$, coordinates as (6,7) are not reached by the blue lines which correspond to our equations. To reach those values, we need to translate equations on x-axis by 1 (green line): $y = x + (x_y - (x_p + 1))$. For $a = 2$, we need to offset x_p by 1 and 2.

The number of offsets applied to equations in order to reach all the coordinates uniformly with the same coefficient a is determined by the integer value of a:

$$\forall u \in \mathbb{C}_f,\ \sigma(u) = \begin{cases} u & \text{if } u \in \mathbb{Z} \\ \frac{1}{u} & \text{otherwise} \end{cases}. \tag{7}$$

By using $\sigma(a)$, we recover the number of needed offsets to represent all values in $\{-8, ..., 7\} \times \{-8, ..., 7\}$ by the coefficient a. All these equations can be represented in the set θ of equations:

$$\theta = \{\forall x, \forall y \in \{-8, ..., 7\}, \forall a \in \mathbb{C}_f, \exists x_p \in \{-8, ..., 7\}, \exists l \in [0, \sigma(a)] \mid y = a \times x + -\text{sgn}(a) \times x_p - a \times (x_p + l)\}.$$

As explained previously, these equations need to be represented by using only 1 byte with 4 bits which are allocated to represent the 16 values of the interval $\{-8, ..., 7\}$ as coefficient b. While the 4 remaining bits predefine which coefficient a and value of offset l are used as shown in Table I.

a	l
0	0
1	0
1	1
1/2	0
1/2	1
1/2	2
2	0
2	1

a	l
2	2
0*	0
-1	0
-1	1
-1/2	0
-1/2	1
-1/2	2
-2	0

Table I
TABLE OF EQUATION a COEFFICIENTS AND l OFFSETS FOR A MAXIMUM OF 256 EQUATIONS. (0* MEANS A VERTICAL LINE)

To prevent pixel expansion, only 256 equations are used. This restriction reduces the number of usable lines and bind our scheme to a maximum of seven shares. This means that our method is a (2, n) threshold scheme where $n \in \{2, ..., 7\}$.

C. Permutation stream shuffle

After the generation of the shares, a supplementary step is used in order to increase the statistical robustness of our scheme. Like the diffusion phase in Section III-A, this step is designed to protect against histogram analysis. Each time a pixel is processed, a permutation table noted T_i is defined to change the pixel values used as equations noted $E_j(i)$ where $j \in \{1, ..., n\}$ and is the index of a share:

$$T_i = \text{Shuffle}(T_{i-1}, p(i-1)). \tag{8}$$

Each time the method shares a new pixel of the secret image it shuffles the previous permutation with a pseudo-random function by using the previous pixel value $I(i-1)$ as key to create a new permutation. This approach is similar to a block cipher mode such Output Feedback (OFB). For the first pixel, the permutation P_0 is initialised to $(0, ..., 255)$.

After the method has selected a grey-level value acting as a line equation noted $E_j(i)$ where j is the current generated

share, the value associated with the permutation table T_i becomes the pixel value of the pixel in the final share $S_j(i)$:

$$S_j(i) = T_i[E_j(i)]. \quad (9)$$

D. Secret recovery

The recovery phase consists in reconstructing a secret image from, at least, two shares. Quite similar to the encryption phase, the recovery algorithm works as follows:

Step 1: Recover the pixels at the same position;
Step 2: Change/Permute the pixel values using the permutation table T_i;
Step 3: Convert the value of each pixel into line equations;
Step 4: Identify the intersection point (x, y) by solving the linear system of line equations from Step 2;
Step 5: Recover the pixel value $p_{i_D}(i)$ using the coordinates (x, y) of the intersection point;
Step 6: Use the inverse transformation of the diffusion phase on the value from Step 5;
Step 7: Store the result from Step 6 as a pixel in the reconstructed secret image;
Step 8: Update the permutation table T_{i+1} using the value of pixel $p(i)$ as shuffling key;
Step 9: Perform Steps 1-8 for all pixels;

IV. Experimental Results

This section presents experimental results of the proposed secret image sharing method. Fig. 7 illustrates the application of our scheme by using a threshold (2, 3) on the image "Lena". (2, 3) means that we generate three shares and the secret image can be recovered by grouping at least two shares. Fig. 7.a is the secret grey-level image, Fig. 7.f is the reconstructed secret image and they are exactly the same. Fig. 7.b is the diffused image of Fig. 7.a. Fig. 7.c-e show the three generated shared images.

A. Statistical analysis

Thanks to our method, the secret image is recovered without loss of information. As shown in Table II, vertical and horizontal correlations fall almost to zero for the share. Also, the entropy increases to nearly reach the maximum of 8 bits per pixel.

	PSNR (in dB)	Entropy (in bpp)	H.Corr.	V.Corr.
Share	9.04	7,99	0.00396	−0.00968
Reconstructed Secret Image	∞	7,40	0.96616	0.98579

Table II
Statistics of one share and the reconstructed secret image: PSNR (in dB), Entropy (in bpp), Horizontal and vertical correlations.

These experimental results illustrate three properties; generated shares of our scheme appear naturally as a random noise; encoded equations do not reveal the histogram of the secret image as shown in Fig. 8 thanks to the diffusion phase and the permutation stream shuffle maintaining a high level of entropy;

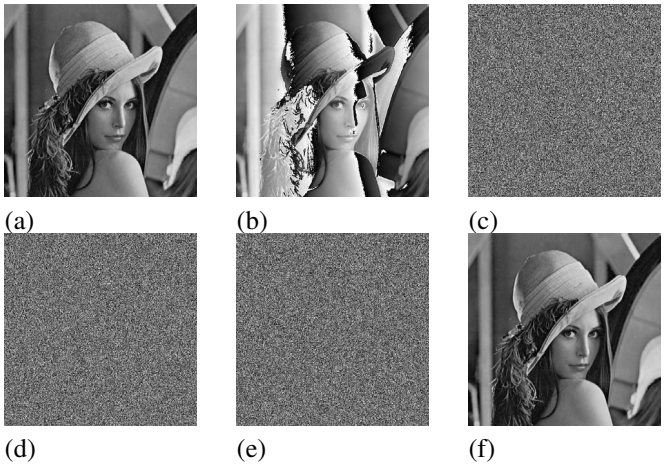

Figure 7. a) The 512 × 512 original secret image, b) The diffused image of (a), (c-e)The 3 512 × 512 shares, f) The reconstructed lossless 512 × 512 secret image from any groups of two shares.

finally, each pixel is shared independently which increases the security of our scheme.

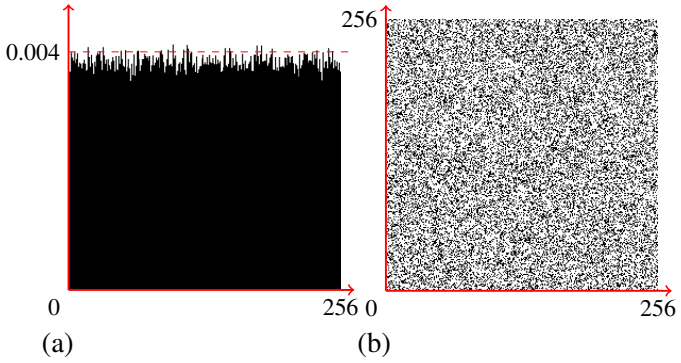

Figure 8. Statistical analysis of the share at Fig.7.c: a) Probability density function of pixel value, b) horizontal and vertical correlation.

In order to validate our results, we applied our method on the 10.000 images from the BOWS-2[1]database with a threshold (2, 7) and results are presented in Table III.

	PSNR (in dB)	Entropy (in bpp)	H.Corr.	V.Corr.
BOWS-2	8.143 (±.897)	7.997 (±.094)	0.00018 (±.0141)	−0.00026 (±.0141)

Table III
Mean values and standard deviations for PSNR (in dB), entropy (in bpp), horizontal and vertical correlations for shares generated from the BOWS-2 database.

As shown in Table III, generated shares have PSNR less than 10 dB, correlations very near to 0 and entropy close to 8 bits per pixel.

B. Security discussion

An adversary cannot figure out the content of the secret image from only one share even with the weakness of Blak-

[1]BOWS2 Website : http://bows2.ec-lille.fr/

ley's scheme. The probability to find out the secret from a share in our scheme was at first $(\frac{1}{7})^{(512\times512)}$ because of the maximum number of equations available. Since it is a very small probability, attackers could obtain an image by forging shares with brute force. However, with the phases of diffusion and permutation stream shuffle, attackers need to find the right permutation for each pixel to reconstruct the secret image. This means to find the right arrangement of 256 by 256 each time. It decreases the probability to recover the secret by forgery to $\frac{1}{(A_{256}^{256}\times7)^{(512\times512)}}$. So, classic brute force approach becomes far more efficient to recover the secret image than share forgery. So, the guessing probability is bounded to $(\frac{1}{256})^{(512\times512)}$ like a pure Shamir's Scheme using a 256-Galois Field [3]. In comparison Thien and Lin's method, without the permutation, has a guessing probability of $(\frac{1}{251})^{\frac{(512\times512)}{2}}$ because of their small share property [5].

C. Comparison analysis

Table IV illustrates comparisons between our scheme and previous secret sharing schemes [3, 5, 6, 7, 8, 9, 10, 11]. Shares generated by our scheme preserve the size of the secret image and the quality of the reconstructed secret image. Moreover, our scheme proposes a greater robustness than previous work to brute force guessing attacks as we discuss in Section IV-B.

Image Sharing	Share Size	Lossless Recovery	Meaningful Shares	Guessing probability
Shamir [3]	1	×	×	$\frac{1}{251^{(512\times512)}}$
Shamir over GF(256) [3]	1	✓	×	$\frac{1}{256^{(512\times512)}}$
Thien *et al.*, Wu *et al.* [5, 6, 8]	1/2	×/✓	×/✓	$\frac{1}{251^{(512\times512)/2}}$
Lin *et al.* [7]	4	×	✓	$\frac{1}{251^{(512\times512)}}$
Tso [9]	1	×/✓	×	$\frac{1}{51^{(512\times512)/4}}$
Tsai *et al.* [10]	2	✓	×	$\frac{1}{251^{(512\times512)}}$
Chen *et al.* [11]	1	✓	×	$\frac{1}{64^{(512\times512)/2}}$
Our scheme	1	✓	×	$\frac{1}{256^{(512\times512)}}$

Table IV
COMPARISONS WITH PREVIOUS SECRET SHARING SCHEMES FOR A (2, N) THRESHOLD.

Other schemes tend to reduce the size of shares by losing quality as a function of the value of k [5, 6, 7, 8]. So, secret image size has to be a multiple of k. These schemes propose to enhance confidentiality of the secret by using steganography abilities to hide their small shares into meaningful images instead of increasing the robustness [6, 7, 8]. However, since recent work, a pure Shamir's scheme [3] based on a 256-Galois Field instead of a prime finite field can produce the same results as our scheme arbitrary for (k, n). By using quantization over pixel values, Tso reduces also the size of the shares by $\frac{1}{4}$, but at the recovery time, fewer images are reconstructed without loss of information.

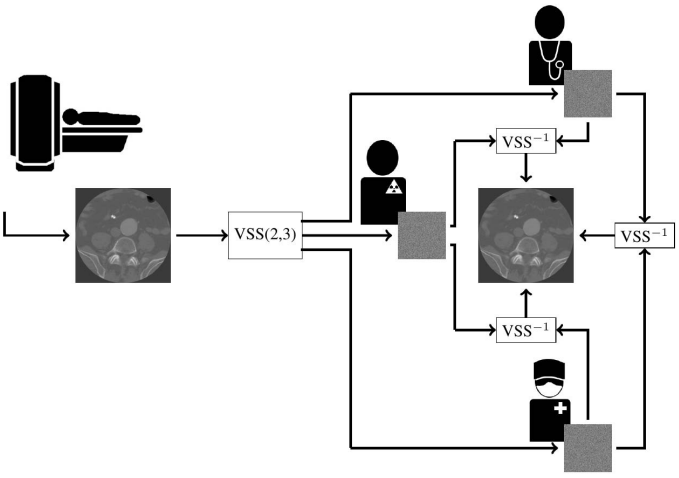

Figure 9. Use case of our scheme for medical purposes. VSS(2,3) is an application of our Visual Secret Sharing scheme using $k = 2$ and $n = 3$. VSS^{-1} is our recovery method.

D. Medical application case

Fig. 9 presents a case of application of our method to secure medical images. After a patient has passed medical examination such a CT scan, the image is securely shared between various practitioners (radiologist, generalist practitioner, surgeon, anaesthetist, *etc*...). And any pairs of practitioner can access to the medical image by bringing their share.

V. CONCLUSION

In this paper, we proposed an efficient (2, n) secret image sharing scheme which is based on linear equations. Our method allows us to share a grey-level image without increasing the size of the shares and to reconstruct it without loss of information. With our approach, n shares are generated and have the same size as the secret image. Pixel values are converted into a two-coordinate system, then in the sharing process, for each pixel we select randomly n line equations going through the associated coordinates. The way we define equations over a simple byte, generate a dense and efficient set of equations intersecting themselves uniformly in a bounded space. Any 2 of them can reconstruct the secret image. Thanks to the diffusion transformation and permutation stream shuffle phases, our scheme proposes a greater robustness to brute force attacks preventing information leaks of the secret by maintaining an entropy similar to classic encryption methods. It is also possible to use it for color image by considering it as 3 grey-level images.

REFERENCES

[1] G. C. Kessler, "An overview of cryptography," 2003.

[2] G. R. Blakley, "Safeguarding cryptographic keys," *Proc. of the National Computer Conference1979*, vol. 48, pp. 313–317, 1979.

[3] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[4] M. Naor and A. Shamir, "Visual cryptography," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1994, pp. 1–12.

[5] C. Thien and J. Lin, "Secret image sharing," *Computers & Graphics*, vol. 26, no. 5, pp. 765–770, 2002.

[6] ——, "An image-sharing method with user-friendly shadow images," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 12, pp. 1161–1169, 2003.

[7] C. Lin and W. Tsai, "Secret image sharing with steganography and authentication," *Journal of Systems and software*, vol. 73, no. 3, pp. 405–414, 2004.

[8] Y. Wu, C. Thien, and J. Lin, "Sharing and hiding secret images with size constraint," *Pattern Recognition*, vol. 37, no. 7, pp. 1377–1385, 2004.

[9] H. Tso, "Sharing secret images using blakley's concept," *Optical Engineering*, vol. 47, no. 7, pp. 077 001–077 001, 2008.

[10] M. Tsai and C. Chen, "A study on secret image sharing," in *Proceedings of the 6th International Workshop on Image Media Quality and its Applications, Tokyo, Japan*. Citeseer, 2013.

[11] C. Chen, W.-Y. Fu, and C. Chen, "A geometry-based secret image sharing approach." *J. Inf. Sci. Eng.*, vol. 24, no. 5, pp. 1567–1577, 2008.