2014 IEEE International Conference on Systems, Man, and Cybernetics
October 5-8, 2014, San Diego, CA, USA

A lossless $(2, 8)$-chaos-based secret image sharing scheme

Long Bao, Yicong Zhou* and C. L. Philip Chen
Department of Computer and Information Science,
University of Macau, Macau, 999078 China
*Email: yicongzhou@umac.mo

***Abstract*—This paper introduces a new $(2, 8)$-secret image sharing scheme integrating the chaos-based image encryption with secret image sharing. It divides the secret image into 8 encrypted shares. Combining any two or more shares is able to completely reconstruct the secret image without any distortion. Each image share is only one pixel larger than the secret image in row and column directions. The proposed scheme is able to directly process the secret images with various formats such as the binary, grayscale, and color images. Experimental and comparison results demonstrate the excellent performance of the proposed scheme.**

***Index Terms*—secret image sharing, chaotic system, image encryption.**

I. Introduction

In the community of data security, secret image sharing is famous because of its special and interesting function. It usually divides a secret image into a number of image shares. A specific combination of those image shares will result in a successful reconstruction of the original secret image. The generation model of secret image sharing is called the (k, n) secret image sharing which generates n different image shares. Only when the number of utilized shares is larger than or equal to k, the successful reconstruction of the original secret image will be achieved. Otherwise, combining less than k image shares yields a noise-like image with no information about the original secret image. Y. Hou proposed a progressive visual cryptography [1] in which a combination of more shares generates a reconstructed image with better visual quality. These properties offer secret image sharing a wide range of applications, such as sharing a secret image among some users.

Visual cryptography (VC), which was firstly introduced by Naor and Shamir [2], is an important and active research topic of secret image sharing. Based on the characteristics of the human visual system, VC can reconstruct the original secret image without using any computer device. Many VC methods have been proposed [3]–[9]. Several VC schemes were proposed with specific properties. In 2009, Shyu et al proposed a VC method to deal with multiple secret images [10]. F. Liu et al proposed a VC scheme [11] to provide cheating prevention. R. Wang [12] and C. Yang [13] have constructed region incrementing VC schemes to deal with a single original image with different security levels. However, VC has its intrinsic disadvantages such as large pixel expansions and poor visual quality of reconstructed results. These limits VC in many real applications. Furthermore, many VC schemes are feasible for only binary images.

To address these VC problems in this paper, we introduce a new chaos-based secret image sharing scheme to deal with various types of secret images, including the binary, gray-scale and color images. It is a combination of the chaos-based image encryption and secret image sharing. Hence, the original secret images can be protected with a high security level and can be completely reconstructed without any data loss. Moreover, compared with traditional VC methods, the proposed scheme generates the image shares with a similar size as the original secret image and thus saves a large amount of storage and transmission costs.

The rest of this paper is presented as follows. The new image sharing scheme will be introduced in detail in Section II. Several experiment results of different secret images, such as binary, gray-scale and color images, will be given in Section III. And the comparison with existing image sharing algorithms are shown in Section IV. Finally, a conclusion will be drawn in Section V.

II. New Secret Image Sharing Scheme

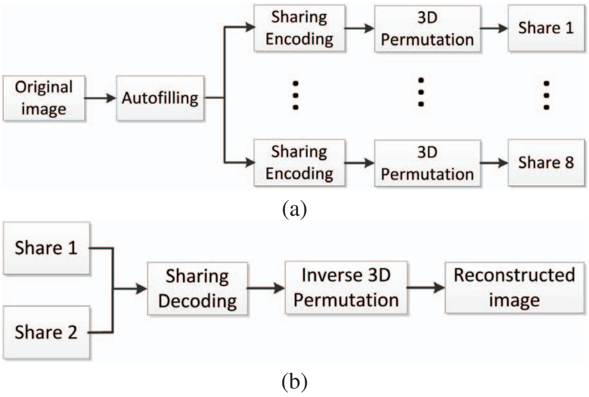


Fig. 1: The proposed $(2, 8)$-chaos-based secret image sharing scheme. (a) The generation phase and (b) the reconstruction phase.

Here, we introduce the new $(2, 8)$-chaos-based secret image sharing scheme (CSISS). Fig. 1 shows its generation phase for creating image shares and its reconstruction phase for recovering the secret image. In the generation phase, the proposed CSISS first uses an autofilling process to add random

978-1-4799-3840-7/14/$31.00 ©2014 IEEE

numbers to the surrounding of the original secret image, encodes the secret image into 8 parts, and performs a 3D permutation them to generate 8 image shares. To reconstruct the secret image, any two image shares are required and applied with the inverse processes of the generation phase to achieve complete reconstruction of original image without any distortion.

A. Generation Phase

The generation phase of the proposed CSISS aims at transforming the original secret image into several noise-like shares. It consists of three steps: auto-filling, sharing encoding and $3D$ permutation, as shown in Fig. 1(a). The original secret image is with a size of $W \times L$ and a data range of $[0, 255]$.

1) Autofilling: The autofilling process first uses a chaotic map as a random generator to produce a chaotic sequence C which has the same data range of the secret image and the length of $2W + 2L + 4$. This random sequence is one-time-used and unpredictable. It is then put in the surrounding of the original secret image to produce a new image (denoted as A) with a size of $(W + 2) \times (L + 2)$ as shown in Fig. 2.

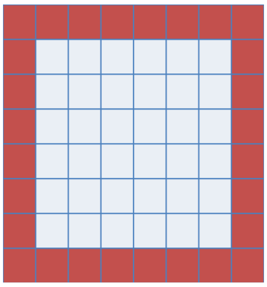


Fig. 2: The image A after auto-filling. The center white region is the original secret image and the red region is the chaotic sequence.

2) Sharing Encoding: The image A is decomposed into 8 bit planes, namely $A_1, A_2, ..., A_8$, where A_i $(1 \leqslant i \leqslant 8)$ is the i^{th} bit-plane. $A = A_1 \parallel A_2 \parallel ... \parallel A_8$ denotes the operation that converts eight bit-planes into a grayscale image.

We then define an image B^j with the same size of the image A, denoting the j^{th} image share to be generated. B^j can be presented as 8 bit planes, namely $B^j = B^j_1 \parallel B^j_2 \parallel ... \parallel B^j_8$. Its i^{th} bit plane B^j_i is defined by Equation (1)

$$B^j_i = \begin{cases} Z_1 & i = j \\ A_i & otherwise \end{cases} \tag{1}$$

where Z_1 is zero matrix with the same of the image A.

3) 3D permutation: Eight bit-planes of each image form a $3D$ binary matrix. The $3D$ permutation is to change all data positions within this binary matrix. As a result, the positions and pixel values are changed. Each image share becomes unrecognized visually. An implementation of the $3D$ permutation is shown in Algorithm 1.

$$x = \mathcal{L}(x, r) = r * x * (1 - x) \tag{2}$$

$$xi = \mathcal{I}(x, m) = (\lfloor x * 10^5 \rfloor \ mod \ m) + 2 \tag{3}$$

$$xi = \mathcal{J}(x) = (\lfloor x * 10 \rfloor \ mod \ 8) \tag{4}$$

$$r = \mathcal{R}(c) = (\frac{(c)_{2 \to 10} + p}{10^{14}} \ mod \ 1)/2.5 + 3.6 \tag{5}$$

$$x = \mathcal{X}(c) = \frac{(c)_{2 \to 10} + p}{10^{14}} \ mod \ 1 \tag{6}$$

where, $(\cdot)_{2 \to 10}$ is to transform the binary data into decimal number. And,

$$p = \sum_{i=1}^{W+2} \sum_{j=1}^{L+2} \sum_{k=1}^{8} (A(i, j)_k)$$

$$\begin{aligned} C3 &= \mathcal{K}(C2) \\ &= C2(1:312) \oplus \cdots C2(312 * d : 312 * (d + 1)) \end{aligned} \tag{7}$$

where,

$$d = \lfloor \frac{2 * (W + L + 2)}{312} \rfloor$$

$$C2 = \mathcal{Z}(C) = [C, Z_2] \tag{8}$$

where, Z_2 is a zero matrix with size of $1 \times (312 - (2 * (W + L + 2) \ mod \ 312))$

Algorithm 1. The 3D permutation

```
Input: $B^j$ with a size of $(W + 2) \times (L + 2)$, and chaotic
    sequence $C$
$E^j = B^j$
$C2 = \mathcal{Z}(C)$
$C3 = \mathcal{K}(C2)$
$r_1 = \mathcal{R}(C3(1:52))$;
$r_2 = \mathcal{R}(C3(53:104))$;
$r_3 = \mathcal{R}(C3(105:156))$;
$x_1 = \mathcal{X}(C3(157:208))$;
$x_2 = \mathcal{X}(C3(209:260))$;
$x_3 = \mathcal{X}(C3(261:312))$;
for $i = 2$ to $W + 1$ do
    for $k = 2$ to $L + 1$ do
        for $z = 1$ to 8 do
            $x_1 = \mathcal{L}(x_1, r_1)$;
            $x_2 = \mathcal{L}(x_2, r_2)$;
            $x_3 = \mathcal{L}(x_3, r_3)$;
            $xi_1 = \mathcal{I}(x_1, W)$;
            $xi_2 = \mathcal{I}(x_2, L)$;
            $xi_3 = \mathcal{J}(x_3)$;
            $a = E^j(i, k)_z$;
            $E^j(i, k)_z = E^j(xi_1, xi_2)_{xi_3}$;
            $E^j(xi_1, xi_2)_{xi_3} = a$;
        end for
    end for
end for
Output: The encrypted image share $E^j$
```

For each position $B^j(i, k)$ in j^{th} image share B^j, it can be presented as $B^j(i, k) = B^j(i, k)_1 \parallel B^j(i, k)_2 \parallel ... \parallel$

$B^j(i,k)_8$. Meanwhile, $\lfloor \cdot \rfloor$ denotes the floor function and "mod" is the module operation. The parameters in parameter matrix is calculated toward the chaotic sequence C and the new image A after auto-filling process.

4) An illustrative example: To make the algorithm easy to understand, Fig. 3 provides an illustrative example to show the result of each step in the proposed CSISS. Here, we apply CSISS to a 3×3 data matrix shown in the top row in Fig. 3. After adding the random data to the surrounding of the data matrix in the auto-filling process, we obtain a 5×5 matrix shown in the second row in Fig. 3. The sharing encoding is then used to generate 8 image shares shown in the third row in Fig. 3. Finally, the 3×3 central part of each image share is encrypted by 3D permutation to obtain the encrypted image share in the bottom row in Fig. 3.

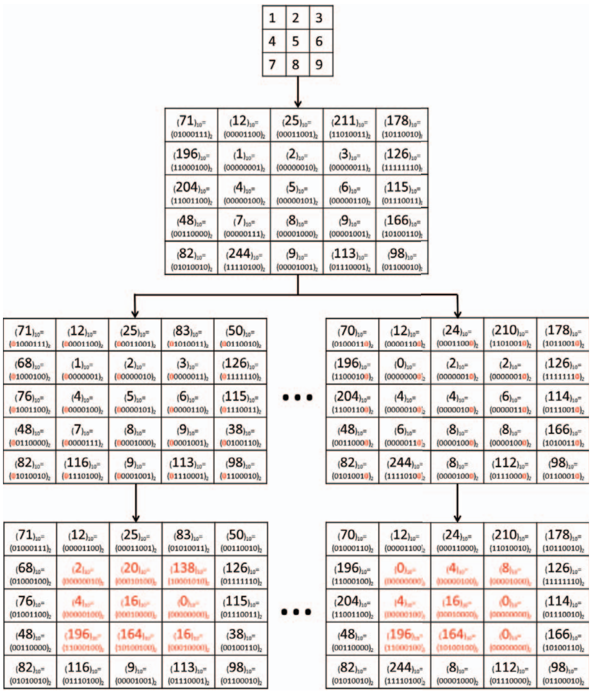


Fig. 3: An illustrative example of the CSISS generation phase. The red fonts indicate the changes in each step.

B. Secret reconstruction phase

Because the proposed CSISS is a $(2, 8)$ scheme, any two shares can reconstruct the original image without any distortion. As shown in Fig. 1(b), the CSISS reconstruction phase is an inverse process of its share generation phase. It consists of two steps: the sharing decoding and inverse 3D permutation.

As defined in Equation (9), the sharing decoding uses two shares E^1 and E^2 with the size of $(W + 2) \times (L + 2)$ to reconstruct the image R.

$$R(i,j) = \begin{cases} E^1(i,j) & \text{if } E^1(i,j) = E^2(i,j) \\ E^1(i,j) + E^2(i,j) & \text{if } E^1(i,j) \neq E^2(i,j) \end{cases} \tag{9}$$

After obtaining the image R, we extract the chaotic sequence according to the inverse process of the auto-filling in Section II-A2. The inverse 3D permutation is used to reconstruct the original image. An illustrative example of the reconstruction phase is shown in Fig. 4.

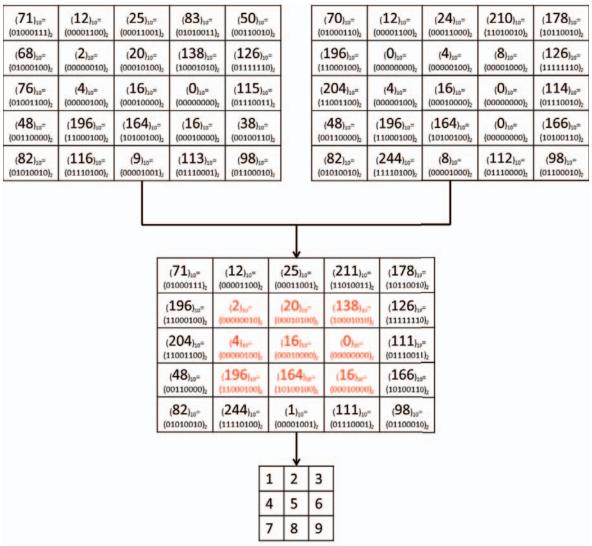


Fig. 4: An illustrative example of the CSISS reconstruction phase. The red fonts indicate the changes in each step.

C. Discussion

The proposed CSISS has at least the following properties:

1) It integrates the secret image sharing with the chaos-based image encryption, achieving a high security level.
2) It is a lossless scheme because the reconstructed image is same as the original secret image. It outperforms than most existing CV methods whose reconstructed original images have low visual quality.
3) It can be used for different types of images, including the binary, gray-scale and color images, while most existing CV methods are designed only for binary images.
4) Each share has a similar size with the original image, significantly reducing the storage and transmission costs.

III. Experiment results

This section provides the simulation results of applying CSISS to the grayscale, binary and color images.

As shown in Fig. 5, we choose a grayscale "child" image as the original secret image. The proposed CSISS is able to transform it into eight different noise-like image shares which are also grayscale images. From these shares, we cannot see any information about the original secret image. In the reconstruction process, each share is unable to reconstruct the original secret image. As can be seen in Fig. 6, the reconstructed results by each individual share are all noise-like images. However, any two or more image shares will reconstruct the original secret image without any distortion, as

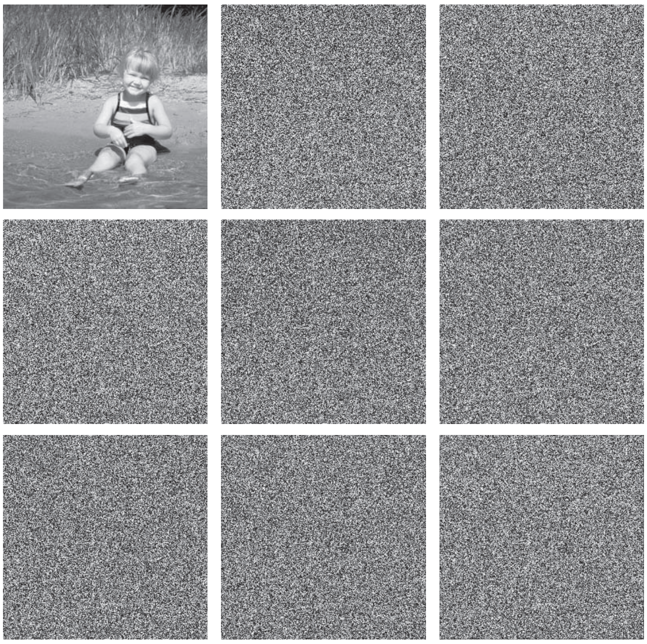
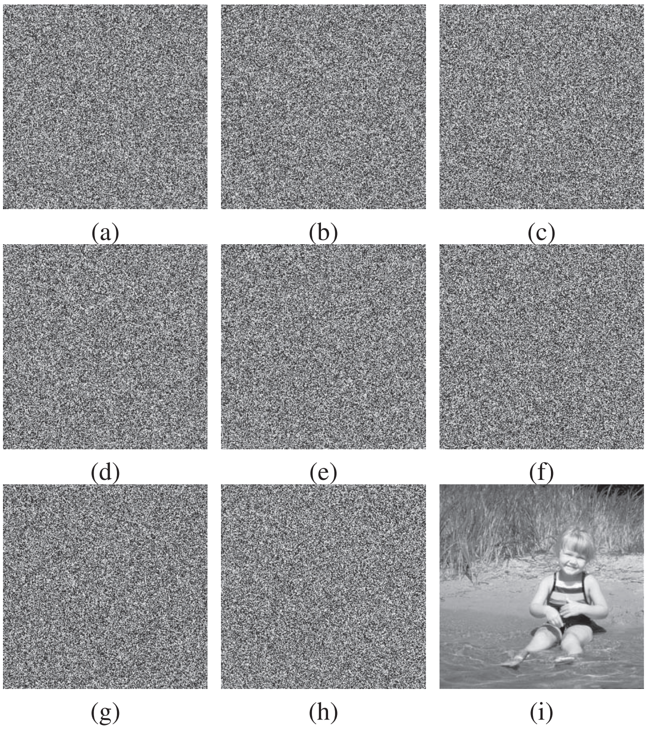


Fig. 5: The original gray-scale image and its eight image shares generated by CSISS.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 6: The reconstructed results using different number of image shares: (a) single Share 1; (b) single Share 2; (c) single Share 3; (d) single Share 4; (e) single Share 5; (f) single Share 6; (g) single Share 7; (h) single Share 8; (i) any two or more shares.

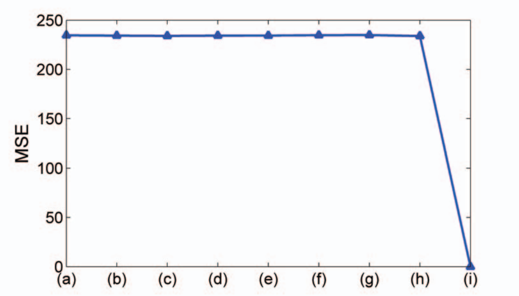


Fig. 7: The MSE results between the reconstructed images in Fig. 6 and the original secret image in Fig. 5.

shown the bottom right image in Fig. 6. Thus, the proposed CSISS is a lossless secret image sharing scheme.

This can be further demonstrated by the quantitative results of the Mean Squared Error (MSE) defined by Equation (10).

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(I_1(i,j) - I_2(i,j))^2 \qquad (10)$$

where I_1 and I_2 are two images with a size of $M \times N$. A larger MSE value means a bigger difference between two images.

We use MSE to measure the differences between the reconstructed images in Fig. 6 and the original secret image in Fig. 5. The MSE results are plotted in Fig. 7. We can observe that the MSE results of the reconstructed images by a single share are extremely high. Only the MSE result of the reconstructed image by any two or more shares is equal to zero, indicating that the reconstructed image is the same as the original secret image. Therefore, the proposed CSISS is able to reconstruct the original secret image without any distortion.

We also apply the proposed CSISS to the binary and color secret images. The experimental results are shown in Figs. 8 and 9. For the binary secret images, we first transform the binary image into a grayscale image by combining eight neighboring binary pixels together to generate a pixel in the grayscale image. The proposed CSISS is then applied to this gray-scale image to generate eight image shares. Finally, converting each pixel in the grayscale image shares into eight neighboring binary pixels yields the corresponding binary image shares as shown in Fig. 8. For the color secret images, we apply the proposed CSISS to each color component individually, combine the corresponding color image shares together to generate the color image shares as shown in Fig. 9. These experiments demonstrate that the proposed CSISS can be successfully applied to images with various formats.

IV. Comparisons

To demonstrate performance of the proposed secret image sharing, we compare it with two existing secret image sharing methods, the Wang's [14] and Thien's [15] algorithms. Table I shows the comparison results including number of shares, Formats of shares, size of shares, MSE values and appearance of shares.

Considering the number and formats of shares, the porposed CSISS and Thien's algorithm are more suitable for real applications because they can generate more shares than the Wang's

	number of Shares	Format of Shares	Size of Shares	MSE values	Appearance of Shares
Wang's algorithm [14]	$n = 2$	binary	N	$MSE = 0$	noise-like image
Thien's algorithm [15]	$n > 2$	gray or binary	N/k	$MSE > 0$	similar to original image
CSISS	$n = 8$	gray or binary	$N/2$	$MSE = 0$	noise-like image

TABLE I: Comparison of different security sharing algorithms

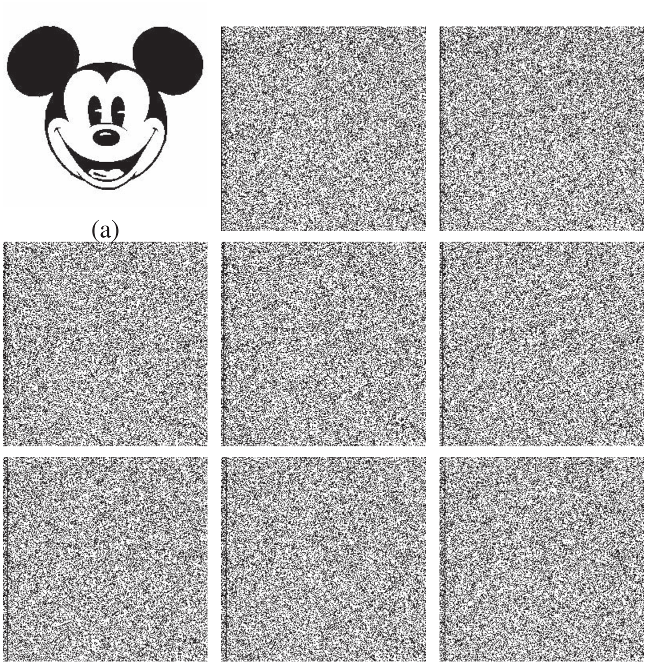


Fig. 8: The original binary image and its eight image shares generated by CSISS.

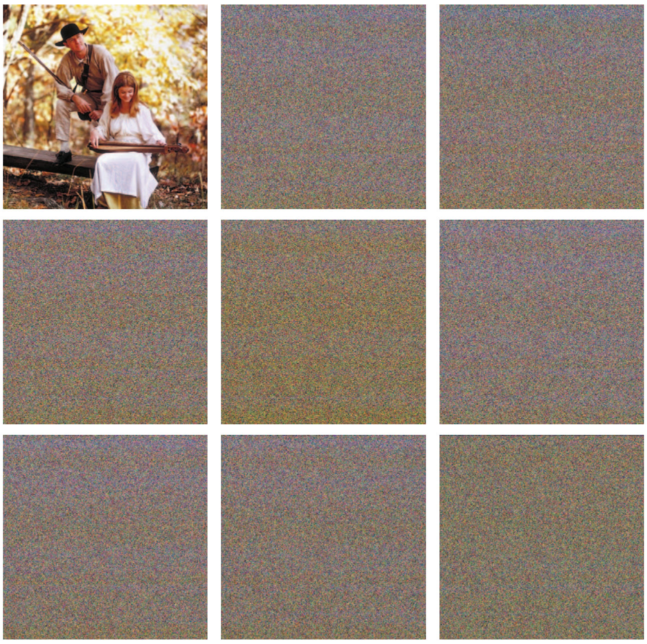


Fig. 9: The original color image and its eight image shares generated by CSISS.

algorithm. Moreover, they can be applied to gray images while the Wang's algorithm can be applied only for binary images. For the size of shares, N denotes the size of the original image. The Thien's algorithm outperforms other two algorithms but it has the data loss problem. As seen from the MSE values between the reconstructed and original images, the proposed CSISS and Wang's algorithm can reconstruct the original image without any data loss while the Thien's algorithm has the data loss in reconstructed images. Furthermore, according to the content of shares, the proposed CSISS and Wang's algorithm generate noise-like shares, indicating that they have a high security level and prevent from information leakage. In summary, the proposed CSISS has better performance than two state-of-the-art methods.

V. Conclusion

In this paper, we introduced a new chaos-based secret image sharing scheme combining the chaos-based image encryption with the secret image sharing. The proposed method is able to protect the original secret image with a high level of security. It can transform the secret images into eight image shares in which any two or more shares are able to completely reconstruct the original secret image without any distortion. The image shares have similar sizes to the original secret images. Experiment results have demonstrated that the proposed method can be applied to different kinds of images, including the binary, grayscale and color images. The comparison results have also demonstrated the excellent performance of proposed CSISS.

Acknowledgement

This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/017/2012/A1 and by the Research Committee at University of Macau under Grants MYRG2014-00003-FST, MRG017/ZYC/2014/FST, MYRG113(Y1-L3)-FST12-ZYC and MRG001/ZYC/2013/FST.

References

[1] Y.-C. Hou and Z.-Y. Quan, "Progressive visual cryptography with unexpanded shares," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 11, pp. 1760–1764, Nov 2011.

[2] M. Naor and A. Shamir, *Visual cryptography*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1995, vol. 950, ch. 1, pp. 1–12.

[3] Y.-S. Lee, B.-J. Wang, and T.-H. Chen, “Quality-improved threshold visual secret sharing scheme by random grids,” *IET Image Processing*, vol. 7, pp. 137–143(6), March 2013.

[4] S.-J. Shyu and M. C. Chen, “Optimum pixel expansions for threshold visual secret sharing schemes,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 960–969, Sept 2011.

[5] F. Liu and C. Wu, “Embedded extended visual cryptography schemes,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 307–322, June 2011.

[6] X. Wu and W. Sun, “Generalized random grid and its applications in visual cryptography,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1541–1553, Sept 2013.

[7] S.-J. Lin and W.-H. Chung, “A probabilistic model of (t, n) visual cryptography scheme with dynamic group,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 197–207, Feb 2012.

[8] R.-Z. Wang and S.-F. Hsu, “Tagged visual cryptography,” *IEEE Signal Processing Letters*, vol. 18, no. 11, pp. 627–630, Nov 2011.

[9] S. J. Shyu, “Visual cryptograms of random grids for general access structures,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 3, pp. 414–424, March 2013.

[10] S. J. Shyu, S.-Y. Huang, Y.-K. Lee, R.-Z. Wang, and K. Chen, “Sharing multiple secrets in visual cryptography,” *Pattern Recognition*, vol. 40, no. 12, pp. 3633 – 3651, 2007.

[11] F. Liu, C. Wu, and X. Lin, “Cheating immune visual cryptography scheme,” *IET Information Security*, vol. 5, no. 1, pp. 51–59, March 2011.

[12] R.-Z. Wang, “Region incrementing visual cryptography,” *IEEE Signal Processing Letters*, vol. 16, no. 8, pp. 659–662, Aug 2009.

[13] C.-N. Yang, H.-W. Shih, C.-C. Wu, and L. Harn, “k out of n region incrementing scheme in visual cryptography,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 5, pp. 799–810, May 2012.

[14] Z. H. Wang, C. C. Chang, H. N. Tu, and M. C. Li, “Sharing a secret image in binary images with verification,” *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 1, pp. 78–90, 2011.

[15] C.-C. Thien and J.-C. Lin, “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, Dec 2003.