The Sixth International Workshop on Image Media Quality and its Applications, IMQA2013
September 12-13, 2013, Tokyo, Japan

A STUDY ON SECRET IMAGE SHARING

Ming-Hong Tsai[1] *and Chaur-Chin Chen*[2]
[1,2]Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan
[2]Institute of Information Systems & Applications, National Tsing Hua University, Hsinchu, Taiwan

ABSTRACT

Information security becomes more and more important while the internet communication grows up. The secret image sharing technique, called (*k, n*) threshold scheme, is a useful method to protect our secret. This technique distributes a secret image to n shadow images preserved by n participants, respectively and we can only reveal the secret image by collecting at least *k* out of *n* shadow images. Fewer than *k* shadow images would not be sufficient to reveal the secret image. This paper discusses three commonly used secret sharing methods and provides our implementation results based on the Chinese Remainder Theorem. It takes fewer parameters than the other methods. Although we can only reveal the secret image with the average root mean square error less than 3, an additional image of the size being a quarter of that of the original one can ensure that the original image is completely recovered.

1. INTRODUCTION

Because of the fast growth of internet technology development, our daily lives cannot be separated from the internet. We can get lots of information we want by powerful search engine, share the articles or photos in blogs, and contact friends by social network. Therefore, information security becomes more and more important nowadays. If we do not process or hide our secret information, the information might be stolen by the hackers easily. The image hiding and watermarking are techniques that can increase the security of the secret information. However, there is a drawback that the information is kept in a single information-carrier. If the information-carrier is lost or destroyed by an attacker, the secret information might disappear. Thus motivated, the secret sharing method might be the better technique that not only increases the security but also has an extremely high opportunity of recovering the secret information completely. It aims to distribute the secret information to several shadow information holders, and we can completely recover the desired information by collecting an enough number of multiple carriers. That is, we can recover the secret based on only some portion of shadow information.

The secret sharing scheme was first proposed by Shamir [3] and Blakley [2] in late 1980s. It is also called (k, n) threshold scheme which should meet the following three requirements, where a secret is represented by a positive integer S.

(1) The secret value S is used to generate n shadows (positive integers): D_1, D_2, … , D_n.
(2) Any k or more shadows can be used to reconstruct the secret value S.
(3) Any k-1 shadows or fewer cannot get sufficient information to reveal the secret value S.

Some (*k, n*) threshold schemes proposed by Shamir [3], Blakley [2], and Asmuth [1], which are applied for image sharing can be found in [5,8,7], respectively.

This paper *lists algorithms* and *discusses the implementation* of the aforementioned schemes for sharing secret images based on modulus operations. *We propose a slight modification based on Ulutas et al.* [1,7] *for an alternative selection of parameters.*

The rest of this paper is organized as follows. Section 2 reviews three algorithms in details for secret image sharing. Section 3 proposes a new scheme for image sharing. Section 4 illustrates the experimental results. The conclusion is drawn in Section 5.

2. ALGORITHMS FOR SECRET IMAGE SHARING：A REVIEW

The secret image sharing is a technique based on secret message sharing [10]. Each pixel value of a secret image is a secret message. We use the secret image to generate shadow images and only require part of these shadow images to reconstruct the secret image. The shadow images should not reveal any information about the image itself such as *silhouette* and *smooth regions*. Ideally, each shadow image should look like random noise so that anyone without the permission won't be able to get any information about a secret image. Some algorithms of image sharing are reviewed as follows.

2.1. Shamir-Based Method [3]

2.1.1. Image sharing algorithm (over n participants)

(1) Suppress all pixels whose gray values greater than 250 to 250.
(2) Permute the pixels of the secret image to get a permuted image, and pick up an integer threshold *k*.
(3) Sequentially take *k* not-shared-yet pixels of the permuted image to form a (*k-1*)-degree polynomial.

$$q(x) \equiv (a_0+a_1x+\ldots+a_{k-1}x^{k-1}) \mod (p=251) \quad (1)$$

where $a_0, a_1, \ldots, a_{k-1}$ are the k pixel values we take. Then generate n pixel values $q(r_1), q(r_2), \ldots, q(r_n)$ for n shadow images, where $1 \leq r_1 < r_2 \ldots < r_n \leq 250$.

(4) Repeat step (3) until all pixels of the permuted image are obtained.

2.1.2. Revealing algorithm

(1) Collect at least k shadow images.
(2) Take the first unused pixel from each of k shadow images.
(3) Use these k pixel values and Lagrange's polynomial interpolation [9] to solve for the coefficients $a_0, a_1, \ldots, a_{k-1}$. Put these coefficients into the permuted image sequentially.
(4) Repeat steps (2) and (3) until all pixels of the k shadow images processed.
(5) Inversely permute the pixels of the permuted image to recover the original secret image.

This method may reveal a distortion secret image because we suppress the gray value greater than 250 to 250. Thien and Lin [5] proposed a special process by transforming the permuted image to an array E to preserve the complete information but it might increase the shadow image size. The sharing and revealing algorithms are given below.

2.1.3. Lossless image sharing algorithm

(1) Pick up a threshold k and permute the pixels of the secret image.
(2) (Transform the permuted image to an array E) Sequentially read in gray value p_i of the permuted image. If $p_i < 250$, then store p_i in E. If $p_i \geq 250$, store a 250 followed by p_i-250 in the array E.
(3) Sequentially take k not-shared-yet elements of E to form a (k-1)-degree polynomial Eq. (3). Then generate n pixels $q(r_1), q(r_2), \ldots, q(r_n)$ for n shadow images, where $1 < r_1 < r_2 \ldots < r_n < 250$.
(4) Repeat step (3) until all pixels of the permuted image are obtained.

2.1.4. Lossless revealing algorithm

(1) Collect any k shadow images.
(2) Take the first unused pixel from each of k shadow images.
(3) Use these k pixels and Lagrange's polynomial interpolation [9] to solve for the coefficients $a_0, a_1, \ldots, a_{k-1}$, and store them in E sequentially.
(4) Repeat steps (2) and (3) until all pixels of the k shadow images have been processed.
(5) (Transform the array E to the permuted image) Sequentially read in the element e_i of E. If $e_i < 250$, then store e_i as a pixel of the permuted image. If $e_i \geq 250$, then read in e_{i+1} and store the value (e_i+e_{i+1}) as a pixel of the permuted image.
(6) Inversely permute the pixels of the permuted image to recover the secret image.

2.2. Blakley-Based Method [2]

2.2.1. Image sharing algorithm (over n participants)

(1) Pick up a threshold k and a parameter d < 251.
(2) Suppress all pixel values larger than 250 to 250.
(3) Randomly generate k-1 parameters $\{0 < g_0, g_1, \ldots, g_{k-2} < 251\}$ and take k pixel values $\{a_0, a_1, \ldots, a_{k-1}\}$ in a lexicographic order from a secret image. Calculate

$$y \equiv (g_0a_0+g_1a_1+\ldots+g_{k-2}a_{k-2}+da_{k-1}) \mod 251 \quad (2)$$

Save the data sequence $\{y, g_0, g_1, \ldots, g_{k-2}\}$ to a shadow image.
(4) Repeat step (2) n times until each shadow image has the data sequence.

Note that: $\{g_0, g_1, \ldots, g_{k-2}, d\}$ should be selected such that every k shadow images can have only one homogeneous solution to recover $\{a_0, a_1, \ldots, a_{k-1}\}$.
(5) Repeat steps (2) and (3) until all pixels of the secret image have been processed.

2.2.2. Revealing algorithm

(1) Take the first unused k pixels from each of any k shadow images, then we can get k polynomials which have k unknown parameters. Solve the simultaneous equations for the data sequence $\{a_0, a_1, \ldots, a_{k-1}\}$.
(2) Repeat step (1) until all pixels of the k shadow images have been processed.

2.3. Asmuth-Based Method [1,7]

2.3.1 Image sharing algorithm (over n participants)

(1) Create a set of integers $\{m_0, m_1, m_2, \ldots, m_n\}$ which satisfies $127 < m_0 < m_1 < m_2 < \ldots < m_n < 257$ and the following two more conditions.
   (i) $\gcd(m_i, m_j) = 1$, for $0 \leq i < j \leq n$
   (ii) $M = \prod_{i=1}^{k} m_i > m_0 \times \prod_{i=1}^{k-1} m_{n-i+1}$
(2) Sequentially take a pixel with gray value p from the secret image.
   If $p < m_0$, compute

$$y = p+r \times m_0 \quad (3)$$

   else compute

$$y = p-m_0+r \times m_0 \quad (4)$$

   where r is a random positive number and $t < r < [(M/m_0)-1]$ in Eq.(3) and $0 \leq r \leq t$ in Eq.(4), where t is a user-specified integer. Note this avoids the same pixel value is recorded as the same integer in a shadow image in this scheme.
(3) Calculate

$$y_i \equiv y \pmod{m_i}, \text{ for } i=1, 2, \ldots, n \quad (5)$$

where y_i is held in the *i-th* shadow image, $1 \leq i \leq n$.

(4) Repeat steps (2) and (3) until all pixels of the secret image are processed, where m_i should be associated with the *i-th* shadow image, $1 \leq i \leq n$.

2.3.2 Revealing algorithm

(1) Collect any *k* shadow images, and sequentially take the first unused pixels y_i for i=1, 2, …, *k*. Apply the Chinese Remainder Theorem [4] based on *k* pairs $\{(m_i, y_i), 1 \leq i \leq k\}$ to reconstruct the corresponding value y.

(2) Compute

$$r = \left\lfloor \frac{y}{m_0} \right\rfloor \quad (6)$$

If $r \leq t$, store the $m_0+(y \bmod m_0)$, otherwise store the value of $(y \bmod m_0)$.

(3) Repeat steps (1) and (2) until all pixels of *k* shadow images are processed.

3. PROPOSED SECRET IMAGE SHARING METHOD

This section introduces our method to implement the algorithm based on Ulutas et al. [7]. We use the leftmost six bits of each pixel to generate *n* shadow images and preserve the rightmost two bits to form an image necessary for a complete recovery. The sharing and revealing algorithms are given as follows.

3.1. The Proposed Secret Image Sharing Method

(1) Create a set of integers $\{m_0, m_1, m_2, \ldots, m_n\}$ which satisfies $63 < m_0 < m_1 < m_2 < \ldots < m_n < 128$ and the following two more conditions.

(i) $\gcd(m_i, m_j) = 1$, for $0 \leq i < j \leq n$.

(ii) $M = \prod_{i=1}^{k} m_i > m_0 \times \prod_{i=1}^{k-1} m_{n-i+1}$

(2) Sequentially take a pixel value *p* of a secret image and compute

$$y = \left\lfloor \frac{p}{4} \right\rfloor + r * m_0 \quad (7)$$

$\left\lfloor \frac{p}{4} \right\rfloor$ means the leftmost six bits of the gray value p and r is randomly chosen between 0 and $[(M/m_0)-1]$.

(3) Keep the least significant two bits of *p* in an array E.

(4) Calculate

$$y_i \equiv y \pmod{m_i}, \text{ for } i=1, 2, \ldots, n \quad (8)$$

where $2y_i$ is kept in the *i-th* shadow image for i=1, 2,…, n, our scheme forces y_i to be saved in the leftmost seven bits.

(5) Repeat steps (2), (3) and (4) until all pixels of the secret image are processed.

(6) Sequentially read 4 elements each time from the array E and make up a pixel to provide necessary image until all elements of E have been processed.

3.2. The Proposed Secret Image Revealing Method [6]

(1) Sequentially read two bits from the necessary image and store them in an array E until all bits are processed.

(2) Collect any *k* shadow images, and sequentially take the first unused pixel to get its leftmost seven bits as shadow message y_i for $1 \leq i \leq k$. Then we can get k pairs $\{(m_i, y_i), 1 \leq i \leq k\}$. Use the Chinese Remainder Theorem to resolve the y value.

(3) Take the first unused element *e* of E.

(4) Reconstruct a pixel value *p* of the secret image by

$$p = (y \bmod m_0)*4+e \quad (9)$$

(5) Repeat steps (2), (3) and (4) until all pixels of *k* shadow images are processed.

3.3. Discussion

Compared with the work of Ulutas et al. [7], our proposed method *does not need to pick up a parameter t* but requires *a necessary image whose size is a quarter of the original image* to ensure the secret image can be completely recovered. However, if we do not have this necessary image, we still can reveal an image, close to the original one (with the root mean square error less than 3). Figure 7 demonstrates this effect.

In our scheme, every pixel of each shadow image uses only seven bits, therefore, we can use the rightmost bits of all pixels to save some other information such as $\{m_i$'s$\}$ in each shadow image. The proposed method does not require *a threshold parameter t*. It only needs to memorize m_0 and *(k, n)*-threshold additionally.

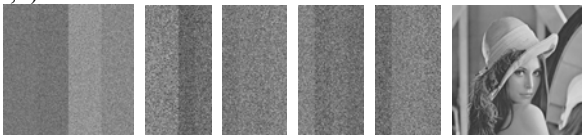
4. EXPERIMENTAL RESULTS

Experimental results of the three existing secret image sharing methods and our proposed secret image sharing method are demonstrated in this section. A *(k,n)*=(2,4)-threshold scheme is implemented. The parameters we choose to implement associated with the generated shadow images will be given. Figure 1 shows the *testl secret image Lenna* with the size 512x512.

Figure 1. The 512x512 secret image Lenna.

4.1. Results of Shamir-Based Method

(2,4) threshold scheme



(a) (b) (c) (d) (e) (f)

Figure 2. (a) The permuted secret image, (b)~(e) are 4 shadow images, (f) the revealed image by (b) and (c).

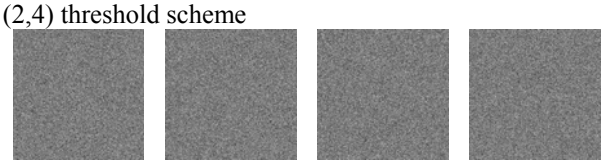
This test secret image Lenna does not contain pixels whose gray values larger than 250, so the size of shadow image is *1/k* of the size of the secret image. The size of each shadow image is 512×256. We choose r_1=1, r_2=2, r_3=3, r_4=4, to generate the shadow images. On the other hand, Figure 3 shows the results of shadow image generated directly from the image Lenna. Some contents of the image Lenna can be seen. *We suggest that the secret image pixels be shuffled before those shadow images are generated.*

(a) (b) (c) (d)

Figure 3. (a)~(d) the shadow images without permuting the secret image.

4.2. Results of Blakley-Based Method

(2,4) threshold scheme



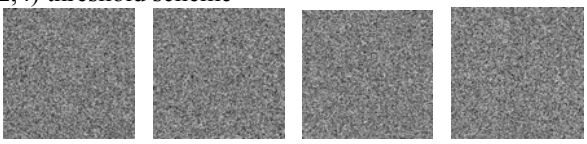
(a) (b) (c) (d)

Figure 4. The Shadow images look like random noise.

We choose d = 14 in Eq.(2). This method cost much more time to randomly select the *n* parameter sets $\{g_0, g_1, \ldots, g_{k-2}\}$ such that the coefficients of any *k* sets of parameters given in the linear system of equations in (2) can lead to a unique solution.

4.3. Asmuth-Based Method [1,7]

(2,4) threshold scheme



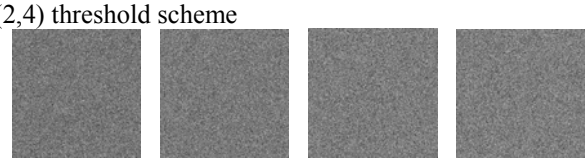
(a) (b) (c) (d)

Figure 5. The shadow images look like random noise.

We choose $m_0 = 247$, $m_1 = 251$, $m_2 = 253$, $m_3 = 255$, $m_4 = 256$, and ***t = 100***. This set meets the two conditions given in section 2.3.1. (1)(i, ii).

4.4. Proposed Secret Image Sharing Method [6]

(2,4) threshold scheme



(a) (b) (c) (d)

(e)

Figure 6. (a~d) The shadow images, (e) a *necessary image* consisting of two least significant bits of Lenna.

(a) (b)

Figure 7. (a) The test secret image, (b) The revealed image by ignoring the two least significant bits.

We choose $m_0 = 119$, $m_1 = 123$, $m_2 = 125$, $m_3 = 127$, $m_4 = 128$. Figure 7(b) shows the result without using the *necessary image*. The PSNR = 42.70 looks visually very close to Figure 7(a).

4.5. Runtime of Different Methods

The runtimes in a contemporary PC with dual core processors 5200+ running Windows XP are summarized in Table 1. Shamir-Based method is the fastest one in the sharing part and Asmuth-Based method is the fastest one in the revealing part on a 512x512 image.

Table 1. Runtime (in sec.) on a 512x512 image Lenna.

Methods	Sharing time(s)	Revealing time(s)
Shamir-Based	0.2869	0.2498
Blakley-Based	6.9204	1.2598
Asmuth-Based	1.3837	0.0886
Proposed One	1.2377	0.0685

5. CONCLUSION

This paper reviews and implements three secret image sharing algorithms [10]. The Shamir-based method utilizes a scheme of Lagrange polynomial interpolation [9] to recover the secret image from *k out of n* shadow image generated from an original secret image. This method might expose some image features in the shadow images without preprocessing the secret image. The Blakley-based method requests the least additional parameters and the shadow images do not need to take the index information. But this method needs to find *n* hyper-planes such that any *k* of out of *n* hyper-planes should get a unique solution which may cause a longer runtime. The Asmuth-based method is the fastest one in computation. However, due to the implementation on 8-bit gray level images, the number of shadow images should not be more than ten because the maximum gray value of shadow images depends on the increasing sequence of m_i we choose. The proposed method can use the rightmost one bit of each shadow image to carry some information such as index m_i. It requires *an additional necessary image* to ensure a complete recovery of secret image although it can also reveal *an*

image very close image with PSNR value over 38.58 (rms <3).

The size of shadow images in Shamir-Based method [3,5] can be *1/k* for a (*k*,*n*)-threshold scheme. The size of shadow images in the other methods are the same as that of the original secret image. By applying the existing schemes [1-6] directly on a secret image, the silhouette and smooth regions of an image may appear in a shadow image. We propose doing a random permutation on pixels of a secret image before applying any (k-n)-threshold based scheme to overcome this problem.

Yang discusses a visual secret sharing (VSS) scheme [14], the size of shadow images will be expanded. Lin et al. [13] proposes a VSS scheme without expanding a pixel size of shadow images. Ulutas et al. [12] reported a combination of steganography with image sharing for authentication. Thus motivated, our future work will be investigate a combination of schemes of image sharing, steganography, and visual secret sharing for the authentication.

6. ACKNOWLEDGMENTS

We appreciate the encouraging suggestions and great comments from anonymous reviewers. This work is supported in part by the Taiwan Grant NSC 101-2221-E-007-125-MY3.

7. REFERENCES

[1] C. Asmuth and J. Bloom, "A Modular Approach to Key Safeguarding," IEEE Trans. On Information Theory, Vol.29, No.2, 208-210, 1983

[2] G.R. Blakley, "Safeguarding cryptographic keys," Proceedings of the National Computer Conference, New York, *American Federation of Information Proceeding Societies*, Vol. 48, 313-317, 1979.

[3] A. Shamir, "How to share a secret," *Communications of the ACM*, Vol. 22, No.11, 612-613, 1979.

[4] D.R. Stinson, "Cryptography: Theory and Practice," Champman&Hall/CRC Press, 2006.

[5] C.C. Thien and J.C. Lin, "Secret image sharing," *Computer & Graphics*, Vol.26, No.1, 765-7710, 2002.

[6] M. H. Tsai, "A Study on Secret Image Sharing," M.S. Thesis, National Tsing Hua University, Hsinchu, Taiwan, March, 2013.

[7] M. Ulutas, V.V. Nabiyev, and G. Ulutas, "A New Secret Image Sharing Technique Based on Asmuth Bloom's Scheme", *Application of Information and Communication Technologies*, 1-5, 2009.

[8] C.C. Chen, W.Y. Fu, and C.C. Chen, "A Geometry-Based Secret Image Sharing Approach", *Proceedings of Image and Vision Computing*, Dunedin, NZ, 428-431, 2005.

[9] http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html, last access on July 9, 2013.

[10] http://en.wikipedia.org/wiki/Secret_sharing, last access on July 9, 2013.

[11] http://en.wikipedia.org/wiki/Steganography, last access on July 9, 2013.

[12] G. Ulutas, M. Ulutas, and V.V. Nabiyev, "Secret image sharing scheme with adaptive authentication strength", *Pattern Recognition Letters*, Vol.34, No.3, 283-291, 2013.

[13] T.L. Lin, S.J. Horng, K.H. Lee, P.L. Chiu, T.W. Kao, Y.H. Chen, R.S. Run, J.L. Lai, R.J. Chen, "A novel visual secret sharing schemes for multiple secrets without a pixel expansion", *Experts Systems with Applications*, Vol.37, 7858-7869, 2010.

[14] C.N. Yang, "New visual secret sharing schemes using probabilistic method", Paatern Recognition Letters, Vol.25, 481-494, 2004.