

A GEOMETRY-BASED SECRET IMAGE SHARING APPROACH

Chien-Chang Chen¹, Wen-Yin Fu¹, Chaur-Chin Chen²

¹Department of Information Management, Hsuan Chuang University, Hsinchu 300, Taiwan

²Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan

Abstract

This paper solves the visual cryptography problem by the Blakley scheme that is a geometry-based secret sharing strategy. We first partition the protected image into non-overlapping sets of k pixels. Every k pixels forms a point under a k -dimensional space and the set solution to each generated affine hyperplane intersecting in this point stores as one shared image. Gathering k shared images can calculate the reconstructed image that is consistent with the protected image. The experimental results demonstrate that the proposed approach is a simple but efficient secret image sharing approach.

Keywords: Secret sharing, Threshold, Blakley, Visual cryptography

1 Introduction

Nowadays, everyone can reproduce and spread digital image easily. Thus, how to preserve important image secretly is a very important issue. In recent years, secret image sharing is a technique to overcome this problem. Secret image sharing technique generates several shared images from the protected image and we can calculate the reconstructed image by gathering enough number of different shared images.

Research on secret image sharing techniques mainly divides into two categories. One piles up the shared images to obtain the reconstructed image. The other applies numerical processing on shared images to acquire the reconstructed image. This paper proposes a numerical processing based secret image sharing technique.

Naor and Shamir [4] first introduced the secret image sharing problem and they proposed an approach by piling up binary shared images to share binary secret image. Blundo et al. [3] proposed a secret image sharing approach on grey level images.

Thien and Lin [5] applied numerical processing rather than piling up shared images to share a protected image secretly. They first used a mapping key to permute an image into random-like image and then adopted Shamir's [1] secret sharing scheme to generate shared images.

This paper proposes a new numerical processing

based secret image sharing approach. The proposed approach does not require any mapping function to increase the randomness of shared images. Our proposed approach adopts the Blakley's secret sharing scheme [2] to generate shared images. The Blakley's (k,n) secret sharing scheme is a geometry-based approach that any secret depicts a point in a k -dimensional space and each share represents one hyperplane intersecting in this point. The proposed secret image sharing approach contains the following properties:

1. No extra information is required except thresholds (k,n) .
2. The protected image can be reconstructed by gathering any k different shared images.
3. The protected image cannot be reconstructed when gathering less than k shared images.
4. The generated shared images have the same size with the protected image.

This paper is organized as follows. Section 2 reviews the Blakley's secret sharing scheme. Section 3 illustrates the proposed geometry-based secret sharing approach that includes the sharing algorithm and the recovering algorithm. Section 4 demonstrates the experimental results of the proposed approach. Brief conclusion is given in section 5.

2 Review of Blakley's secret sharing scheme

A secret sharing scheme shares a message among n trustees and any k of them can recover the secret message. Our proposed approach uses Blakley's secret sharing scheme to share protected images. Thus, we briefly introduce this scheme in this section.

Blakley used geometry to solve the secret sharing problem. The secret message is a point in a k -dimensional space and n shares are affine hyperplanes that intersect in this point. The set solution $x=(x_1, x_2, \dots, x_k)$ to an equation $a_1x_1+a_2x_2+\dots+a_kx_k=b$ forms an affine hyperplane. The secret, the intersection point, is obtained by finding the intersection of any k of these planes.

Figure 1 shows an example of (k,n) Blakley's scheme with $k=2$ as a two-dimensional plane and $n=3$ as three shares. The secret is a 2-dimensional point T and three shares (L_1, L_2, L_3) are lines with different parameters (a_1, a_2, b) passing through the point T . When gathering any two lines, for example L_1 and L_2 , we can acquire the secret T by finding the point they intersect.

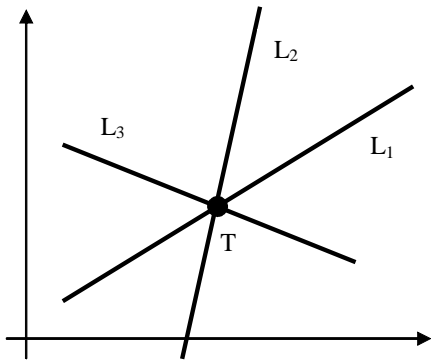


Figure 1: A two-dimensional Blakley's secret sharing scheme

3 The Proposed Secret Image Sharing Approach

A secret image sharing approach includes two algorithms: a sharing algorithm generates shared images from a protected image and a recovering algorithm calculates the reconstructed image from shared images. Section 3.1 introduces the proposed sharing algorithm and section 3.2 introduces the proposed recovering algorithm.

3.1 The sharing algorithm

This section introduces how to share protected image secretly by the Blakley's secret sharing scheme. When applying the Blakley's multi-dimensional scheme to the visual cryptography problem, we should first select two parameters k and n representing

the required shared images to recover the secret and the generated shared images, respectively. Then we partition an image into non-overlapping sets of k pixels. Each k pixels can be a k -dimensional point and each generated shared image represents one affine hyperplane across this point. Assume we want to generate n shared images S_1, S_2, \dots, S_n from the protected image S and gathering k or more shared images can recovery the protected image S . The sharing algorithm is depicted in Figure 2 and illustrated as follows.

1. Select thresholds (k, n) .
2. Partition the protected image into non-overlapping sets of k pixels.
 - 3.1 These k pixels form a k -dimensional point $x=(x_1, x_2, \dots, x_k)$.
 - 3.2 Choose n different solution sets $(a_1, a_2, \dots, a_k, b)$ such that equation $a_1x_1+a_2x_2+\dots+a_kx_k=b$ is satisfied.
 - 3.3 Adjust each set of parameters $(a_1, a_2, \dots, a_k, b)$ to pre-defined bits.
4. Store each set of bits as one shared image.

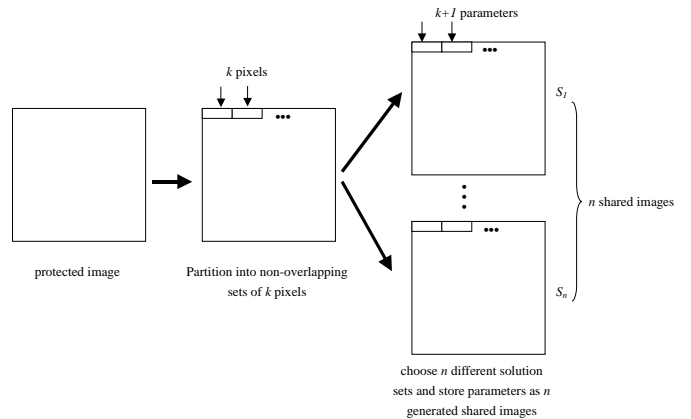


Figure 2: The proposed sharing algorithm

In this proposed sharing algorithm, each set solution in step 3.2 is adjusted to pre-defined bits in step 3.3 and stored into one shared image as shown in step 4. The assignment of pre-defined bits decides the size of each shared image.

We apply the following assignment to adjust shared images with the same size as the protected image. In a (k,n) thresholds, each set contains k pixels (bytes). The generated k parameters (a_1, a_2, \dots, a_k) and one constant parameter b are stored using these k bytes to keep image size invariant. Thus, we have k bytes to store these $k+1$ parameters. We use the following assignment to satisfy this requirement. Each parameter in (a_1, a_2, \dots, a_k) is stored as $\left\lfloor \frac{k \times 8}{k+1} \right\rfloor$ bits and

parameter b is stored as $8 \times k - \left\lfloor \frac{k \times 8}{k+1} \right\rfloor \times k$ bits. The parameters in step 3.3 are adjusted to satisfy this representation.

For example, our experiments select thresholds $(k, n) = (3, 5)$. Thus, the image pixels are partitioned into sets of 3 pixels. From our previous description, we have $\left\lfloor \frac{k \times 8}{k+1} \right\rfloor = 6$ and $8 \times k - \left\lfloor \frac{k \times 8}{k+1} \right\rfloor \times k = 6$. Consequently, we utilize 6 bits to represent each parameter of (a_1, a_2, a_3) and utilize 6 bits to represent parameter b . 6 bits can represent 64 different values and we assign them from -32 to 31 . Thus, parameters (a_1, a_2, a_3, b) are all restricted from -32 to 31 .

3.2 The recovering algorithm

The recovering algorithm calculates the reconstructed image from k or more shared images. When recovering the reconstructed image, we should possess the same thresholds (k, n) as we used in the sharing algorithm. The recovering algorithm is depicted in Figure 3 and illustrated as follows.

1. Adopt the same thresholds (k, n) and k shared images.
2. Partition each shared image into non-overlapping sets of k pixels.
3. Acquire $k+1$ parameters $(a_1, a_2, \dots, a_k, b)$ from k pixels by the method discussed in section 3.1.
4. Find the intersection point of these k hyperplanes constructed by parameters calculated in step 3.
5. Store the coordinate of the intersection point in a k -dimensional space as k pixels in the reconstructed image.
6. Perform all sets through step 2 to step 5 to acquire the reconstructed image.

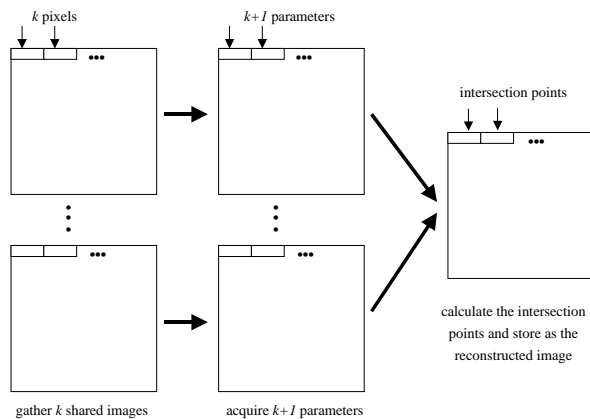


Figure 3: The proposed recovering algorithm

4 Experimental Results

This section demonstrates the experimental results of the proposed geometry-based secret image sharing approach. The fact that the thresholds $(3, 5)$ indicates that we randomly generate 5 shared images and we can calculate the reconstructed image by gathering three ones. The protected image is JET with size 256×256 . Figure 4(a) shows the protected image and Figure 4(g) is the reconstructed image. Both of these two images are consistent. Figures 4(b)-4(f) shows five randomly generated shared images. Experimental results illustrate following three properties.

First, these generated shared images constitute like random noise because of their randomly selected hyperplane parameters. Thus, any adversary can't figure out the protected image from only one shared image.

Second, our proposed approach directly generates randomly shared images. This property is better than Thien and Lin's approach [4] that requires a permutation key to break the protected image into random-like permuted sequence.

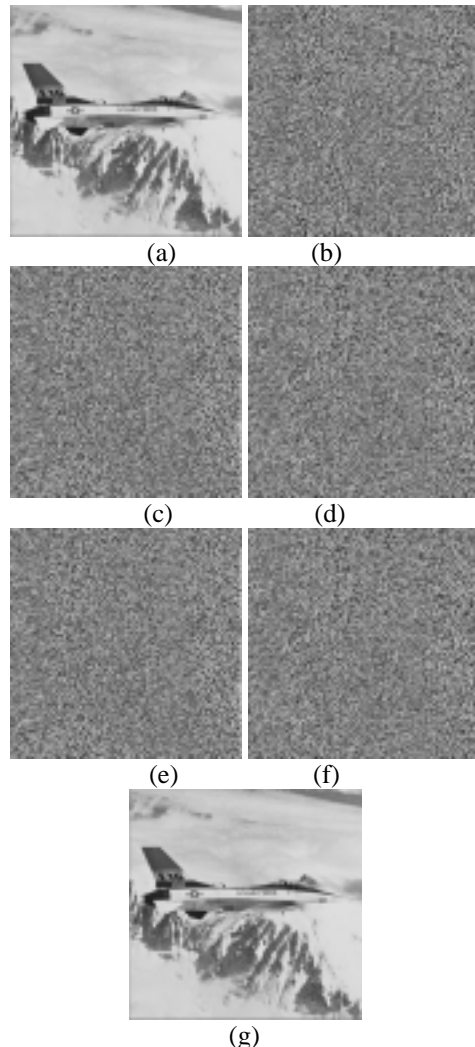


Figure 4: (a) protected image, (b)-(f): five shared images, (g) reconstructed image

At last, we discuss the security of our proposed approach. The proposed recovering algorithm requires k shared images to calculate the reconstructed image. If an adversary gathers $k-1$ shared images, he acquires $k-1$ hyperplanes to calculate the intersection point in a k -dimensional space. His answer will be a line passing through the point. This line contains many solutions but only one is correct. For example, our experimental results of selecting $k=3$ contain over two thousand valid solutions for each hyperplane constituting from 3 pixels. In our example, an image with size 256×256 exists over 20,000 hyperplanes and each hyperplane exists over 2,000 valid solutions which contains over 40,000,000 valid reconstructed images for any brute force attack. This is almost impossible for one to figure out what the protected image is. Thus, experimental results and preliminary analysis illustrate that the proposed approach is a simple but efficient method to share an image secretly.

5 Conclusion

This paper presents a geometry-based secret image sharing approach. The fact that our proposed approach randomly generates all n shared images restricts anyone to recover the reconstructed image when he or she gathers less than k shared images. The fact that our proposed approach does not require an arrangement key also exhibit better characteristic than previous approach.

6 Acknowledgments

Chaur-Chin Chen is supported by NSC Grant 94-2213-E-007-089.

7 References

- [1] Adi Shamir, "How to Share a Secret", *Communication of the ACM*, Vol. 22, pp. 612-613, 1979
- [2] G.R. Blakley, "Safeguarding cryptographic keys", *Proceeding of the National Computer Conference (NCC), 1979, AFIPS Conference Proceedings*, Vol. 48, pp.313-317, 1979
- [3] C. Blundo, A. D. Santis, M. Naor, "Visual Cryptography for grey level images", *Information Processing Letters*, vol. 75, pp.255-259, 2000.
- [4] M. Naor, A. Shamir, "Visual Cryptography", *Advances in Cryptology : Eurpocrypt'94*, Springer-Verlag, Berlin, pp. 1-12, 1995
- [5] Chih-Ching Thien, Ja-Chen Lin, "Secret image sharing", *Computers & Graphics*, Vol. 26, pp.765-770, 2002