Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2009, Article ID 187874, 11 pages
doi:10.1155/2009/187874

Research Article

Improvements in Geometry-Based Secret Image Sharing Approach with Steganography

Mustafa Ulutas, Vasif V. Nabiyev, and Guzin Ulutas

Department of Computer Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey

Correspondence should be addressed to Mustafa Ulutas, ulutas@ieee.org

Received 13 April 2009; Revised 30 September 2009; Accepted 6 November 2009

Recommended by Panos Liatsis

Protection of the sensitive data is an important issue because of the fast development of applications that need exchange of the secret information over the Internet. Secret sharing is an idea proposed by Shamir and Blakley separately with different implementations in 1979. Lin and Tsai proposed a method that uses Steganography to create meaningful shares by using Shamir's secret sharing scheme in 2004. In recent years, researchers work to remove some of the weaknesses of this method. However, all of these methods need cover images four times bigger than the secret image. This arises two problems: increased storage and bandwidth need for shares. We used cover images with the same size as the secret image by using both Blakley's secret sharing approach and Steganography. Therefore, we achieved reduced storage and transmission bandwidth for shares. Besides, the proposed method creates meaningful shares by using Steganography instead of noise-like shares, different from other studies that use Blakley's approach.

1. Introduction

Low cost, fast and reliable networking has increased the data transmitted beyond expectations over the public access network, the Internet. Though it is a public access network, certain data need secrecy during transmission such as military or commercial images to protect them from malicious users. Steganography is a method to protect data from illicit attempts. This method hides a secret message in a cover medium to avoid malicious eyes. The cover could be a digital image, digital video, source code, or even an html code. Attackers do not know that the cover medium has hidden secret data. Therefore, they will not aim to extract the original data. However, a common weakness of this technique is that an entire protected data or image is kept in a single cover medium. The secret data or image cannot be revealed if the medium is lost or corrupted. Secret sharing methods have been used in the literature to overcome this weakness.

Both Shamir and Blakley independently introduced secret sharing schemes in 1979 [1, 2]. These schemes are called (k, n) threshold secret sharing schemes since the secret is distributed among n participants and only k or more participants can recover the secret. The dealer distributes the secret among n participants. Each participant has his or her own piece of secret called share. The secret is revealed, if any k or more of the shares gather.

While Shamir was using polynomial-based technique to share the secret among n participants, Blakley used a geometric approach. Shamir's technique creates a $k-1$ degree polynomial with random coefficients in the range $[0-p)$, where p is a prime number. Secret is the constant term of this polynomial. Evaluated values of y corresponding to different values of x constitute shares. Lagrange's interpolation technique is used for the reconstruction of the secret from any k or more shares. Blakley's technique assumes that secret is a point in a k-dimensional space. Hyperplanes intersecting at this point are used to construct the shares. Coefficients of n different hyperplanes constitute the corresponding n shares.

In these two schemes, a secret is partitioned among n participants. Unless at least k shares are gathered, secret will not be revealed. In other words, any number of shares fewer than k reveal no information about the secret data.

After this pioneering research, Thien and Lin adopted Shamir's polynomial-based secret sharing approach into secret image sharing in 2002 [3]. Then, Wang et al. improved this method by decreasing share images' size [4]. Studies in secret image sharing have focused on different problems of secret image sharing. Dealing with cheater attacks to detect fake shadow, decreasing share images' size, and using color secret image are active topics [5–7].

Thien and Lin proposed a new method for creating meaningful shares which is a shrunken version of the secret image in 2003 [8]. Steganography is used for generation of meaningful shares with secret image sharing approach in 2004 [9]. Data hiding is employed to embed the shares into cover images in this method. Size of the cover images would be twice that of the secret image. Yang et al. proposed a method to overcome some of the weakness in this method in 2007 [10]. Their objective is to prevent the participants from cheating. They improve the authentication ability and qualities of stego images. Chang et al. proposed a method that ensures authenticating the secret image by using the Chinese Remainder Theorem (CRT) in 2008 [11]. Their scheme not only improves the quality of the stego images, but also enhances the authentication ability by using the CRT technique.

All these works reported here have used Steganography with Shamir's secret sharing approach to embed one secret into n meaningful shares.

Although, all these studies used polynomial-based secret sharing approach proposed by Shamir, researchers began to use Blakley's geometry-based secret sharing approach in secret image sharing field recently [12, 13]. Chen et al. and Tso independently adopted Blakley's scheme into secret image sharing area in 2008. Both Shamir's and Blakley's methods have different implementations [12]. While the former was producing share images with the same size of the secret image, the latter claimed to decrease the size of shares [13]. Unfortunately, both methods produce noise-like meaningless shares. Therefore, created shares attract the attention of malicious attackers and threat the security of secret. Meaningful share generation is an important topic in the secret image sharing field.

An enhanced scheme for secret image sharing is proposed in this paper. Both Blakley's secret sharing method and Steganography are used together to share the secret and create meaningful shares. Previous works reported in the literature using Blakley's scheme create noise-like shares, whereas the proposed method creates meaningful shares by using Steganography on the data shared by Blakley's secret sharing scheme. The use of cover

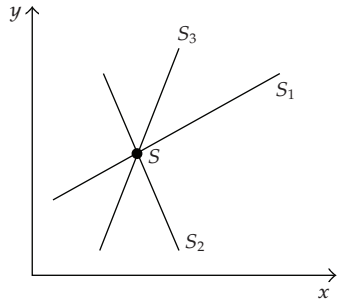

Figure 1: Blakley's secret sharing scheme in two-dimensional space.

images to create meaningful shares in this work both improves the security of the secrets and makes them easy to manage compared to noise-like shares.

While [9–11] based on Shamir's polynomial approach use cover images with twice the size of secret image, cover images in this work based on Blakley's geometric approach are of the same size with the secret image. This is another improvement of our method, which saves both storage space and required time for the transmission of shares.

The remainder of this paper is organized as follows. Blakley's secret sharing approach is briefly described in Section 2. The principles of the proposed method are given in Section 3, and results from this work will be presented in Section 4. Finally, conclusions are given in the last section.

2. Blakley's Secret Sharing Scheme

Blakley's method uses principles of geometry to share the secret [2]. Secret is a point in a k-dimensional space according to this scheme. Affine hyperplanes in this space represent n shares. These hyperplanes intersect at exactly one point in k-dimensional space, which represents the secret.

The set of solution $x = (x_1, x_2, \ldots, x_k)$ to the equation $a_1x_1 + a_2x_2 + \cdots + a_kx_k = B$ forms an affine hyperplane.

Intersection point of any k of these planes represents the secret. Blakley's scheme for two-dimensional space and three shares is shown in Figure 1. The secret is a two-dimensional point S, and three shares are three lines with different parameters (a_1, a_2, B). Three lines represented by S_1, S_2, S_3 intersect at exactly one point; S (secret). Two lines will be sufficient for retrieving the secret, because any two of the three lines intersect at the same point. This simple example demonstrates $(2, n)$ secret sharing scheme.

3. Proposed Method

Secret image sharing algorithm consists of a pair of efficient algorithms. While sharing the secret among n participants with the distribution algorithm constitutes the first stage, second stage gathers any k of shares and reconstructs the secret by using the reconstruction algorithm. The distribution algorithm is executed by a dealer who, given a secret, computes some shares and gives them to n participants. The reconstruction algorithm is executed on

any k or more of the n participants' share to reconstruct the secret. In this section we introduce the proposed distribution and reconstruction algorithms, respectively.

3.1. The Distribution Algorithm

This section introduces the algorithm used to share the secret image among n participants. The dealer chooses a pair of values denoted by (k, n) to distribute the secret image. While n represents the number of participants, k denotes the minimum number of participants that should gather to reveal the secret image. If any k or more participants gather, the secret image will be revealed. Each participant receives his or her own piece called share created by the distribution algorithm in the form of digital images. The algorithm uses steganography to hide the shares into cover images, which renders shares to look like natural images in order to hide them from malicious eyes. The dealer also chooses a set of cover images that look like natural images at the beginning of the algorithm. The number of cover images is equal to the value of n.

The distribution algorithm makes use of Blakley's secret sharing approach to create n shares. Secret is a point in a k-dimensional space according to Blakley's secret sharing scheme. A vector of size $1 \times k$ represents a point also called secret in k-dimensional space. However, the secret is a digital image of size $N \times M$ in the proposed method. Secret image is partitioned in to k pixel groups to overcome this problem. Each pixel group represents a secret point in k-dimensional space.

Coefficients of n different hyperplane equations that intersect at the point where secret is located are distributed to corresponding k subpixel group of n shares. As a result, generated shares' size is equal to the size of the secret image. The details of the proposed distribution algorithm are given below.

The secret image is partitioned into nonoverlapping sets of k pixels. Each k pixel group forms a point in a k-dimensional space, $x = (x_1, x_2, \ldots, x_k)$. Hyperplanes that intersect at this point will be used for the constitution of corresponding shares. k coefficients $(a_1, a_2, \ldots, a_k)$ of the kth degree polynomial and a constant B uniquely define a hyperplane equation. $(a_1, a_2, \ldots, a_k)$ are set to the corresponding cover image's selected subpixel group values. Constant b, satisfying the hyperplane equation in (3.1) at the secret point, is computed and embedded into the same pixel group by using steganography. Slightly modified cover images by B values of corresponding hyperplane equations form corresponding shares at the end of the distribution algorithm. Each share is a slightly modified replica of the corresponding cover image indistinguishable by humans. Human visual system's inability to perceive small contrast change in images is used to modify cover images slightly without noticeable change.

Hyperplane equation used by Blakley's scheme is adopted to support 8-bit gray level image pixel values in the range $[0–2^8–1]$. The equation used during the distribution process is

$$(a_1x_1 + a_2x_2 + \cdots + a_kx_k) \bmod 251 \equiv b. \tag{3.1}$$

It may be shown from (3.1) that ambiguity in the secret recovery will not arise if "modulo 251" is used in computing the value of b. In other words, recovered value of x by solving the linear congruence system during the reconstruction phase might not be unique and might cause a recovery failure unless a prime number is used in the modulo operation.

The value of 251 is the largest prime number less than 256, which represents 8-bit gray level images.

Values of $(a_1, a_2, \ldots, a_k)$ will be selected from corresponding cover image's k pixels. These k pixels will also be used to store the value of B. $\lceil \log_2 251 \rceil = 8$ bit is sufficient to represent the value of B since it cannot exceed 251. Remaining $8 \times k - 8$ bits will be used to compute the values of $(a_1, a_2, \ldots, a_k)$. A simple yet effective method to partition individual share pixels into a_i and b_i portions is given below. The number of Least Significant Bits (LSBs) to store b's lexicographically ordered parts could be determined by

$$B = \left\{ b_i \mid b_i = \left\lceil \frac{l_i}{u_i} \right\rceil, i \in \{0, \ldots, k-1\} \right\}, \tag{3.2}$$

where l_i and u_i are defined as

$$\begin{aligned} U &= \{u_i \mid u_i = k - i, i \in \{0, \ldots, k-1\}\}, \\ L &= \left\{ l_i \mid l_i = l_{i-1} - \left\lceil \frac{l_{i-1}}{u_{i-1}} \right\rceil, i \in \{1, \ldots, k-1\} \right\}, \end{aligned} \tag{3.3}$$

and $l_0 = 8$. Since the number of LSBs to store parts of B is determined, the following equation can be used to compute a_i's from B and cover image subpixel values $(c_1, c_2, \ldots, c_k)$:

$$A = \left\{ a_i \mid a_i = \frac{\left(c_i \wedge \left(256 - 2^{b_{i-1}}\right)\right)}{2^{b_{i-1}}}, i \in \{1, \ldots, k\} \right\}. \tag{3.4}$$

Computation of $(a_1, a_2, \ldots, a_k)$ from corresponding cover image's k subpixels and embedding the calculated value of B into this pixel group are explained in detail in a simple example. $(3, n)$ secret image sharing scheme is used for illustration.

Locations of the bits in each k subpixels at a cover image that will determine the value of $(a_1, a_2, \ldots, a_k)$ and carry the value of B for a $(3, n)$ secret sharing system are shown in Figure 2. Three 8-bit pixel values of the cover image called X, Y, and Z are used for the generation of relative three pixels of a share. The number of bits to be used to compute three values in a hyperplane equation is given in Figure 2 as a_1, a_2 and a_3, respectively. First five bits of X, Y, and Z determine the value of a_1, a_2, a_3, respectively. These values of a_1, a_2, and a_3 will be used to constitute the hyperplane equation as shown below.

$$(a_1 x_1 + a_2 x_2 + a_3 x_3) \bmod 251 \equiv b. \tag{3.5}$$

The value of B defined above can be embedded into least significant bits (LSBs) of X, Y and Z, respectively. Therefore, while most significant bits (MSBs) of X, Y, and Z are used to compute a_1, a_2 and a_3 values of the hyperplane equation, LSB's of X, Y and Z will be used to embed the value of B calculated by using (3.5). The number of bits corresponding to store parts of B is determined by (3.2).

Finally, to prevent illicit attempts that will manipulate cover images, a simple authentication ability is added. The hash function Message Digest 5 (MD5), $H(\cdot)$, with 31-bit

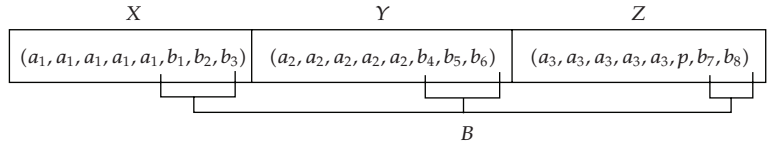

Figure 2: An illustration of the first three pixels that will be used to compute a_1, $a_2$and a_3 and embed parts of B.

input (exclusive the check bit p) is used. 128-bit hash output of the MD5 function is calculated by

$$\begin{aligned} T &= H(X \| Y \| (Z - p)), \\ T &= \{t_i \mid i \in \{1, 2, \ldots, 128\}, t_i \in \{0, 1\}\}. \end{aligned} \tag{3.6}$$

The hash bit p in Figure 2 is calculated by XOR'ed result of 128 bits defined below.

$$p = ((((t_1 \oplus t_2) \oplus t_3) \oplus t_4) \cdots) \oplus t_{128}. \tag{3.7}$$

The method outlined above will be applied for all k subpixel groups of the secret image to create n share. The distribution algorithm to create n shares can be described as the following pseudocode:

Input: a secret image S and n cover images

Output: n stego images.

Step 1. Partition the secret image S and n cover images into k subpixel groups. Repeat the following steps for each k subpixel groups of S.

Step 2. Let k pixels of S represent the point x in k-dimensional space.

Step 3. Form n different hyperplane equations for corresponding k subpixel groups in n cover image by repeating the following steps and return to Step 2.

Step 4. Determine $(a_1, a_2, \ldots, a_k)$ values of a subpixel group in a corresponding cover image using (3.4).

Step 5. Calculate the value of B by using (3.5) and embed this value into relative pixel locations in k subpixels at the corresponding cover image.

Step 6. Calculate the hash bit of this block using (3.7) and embed this bit, p, into correct location.

Feasibility of the proposed method can be illustrated with a simple $(3, 5)$ example. Let the first three pixel values of the secret image be (110 24 72). Five cover images will be partitioned into three pixels subgroups. Therefore, the first three pixels of cover images will be (57 99 220), (85 100 23), (150 100 150), (199 223 223), and (12 33 15), respectively. Each subpixel group of cover images will determine the equation of a hyperplane. These hyperplanes will intersect at exactly (110 24 72), the secret point.

For the first cover image, values of (a_1, a_2, a_3) will be $(7, 12, 27)$, and constituted hyperplane equation will be $7x_1 + 12x_2 + 27x_3 \equiv b \pmod{251}$. If the values of x are substituted in, $7 * 110 + 12 * 24 + 27 * 72 \equiv 241$ is calculated. The binary equivalent of this value $(241 = 11110001_2)$ is embedded into specified bit positions at cover image pixels. Modified pixel values of cover image will then be (63 100 221). Other four cover images' modified pixel values are (80 102 22), (145 100 150), (198 221 220), and (15 33 14), respectively if their pixel values are processed by the same algorithm.

The example given above shows that variation of the intensity values at cover images is limited by three-bit [0–7] range and should not be perceived by malicious attackers.

3.2. Reconstruction Algorithm

The reconstruction algorithm reveals the secret image from k or more share (stego) images. The algorithm can be given in the form of following steps.

Step 1. Use the same (k, n) values and choose any k share (stego) images.

Step 2. Partition each shared image into group of k subpixels.

Step 3. Extract the corresponding subpixel groups of size k from share images. For each subpixel group repeat Steps from 4 to 6.

Step 4. Extract the value of hash bit p from subpixel group.

Step 5. Apply the hash function to the rest of the subpixel groups using (3.4) and compare the result with p. Consistent hash values indicate that the share is authentic and algorithm should proceed to Step 6. Inconsistent hash values should cause the algorithm to output an appropriate error message that the share could be tampered with.

Step 6. Extract the values of $(a_1, a_2, \ldots, a_k)$ and B from current k subpixel group according to Figure 2.

Step 7. A set of k linear independent equations given below is formed as Steps from 4 to 6 are performed. Solve this system of linear congruence using Matrix Inversion technique.

$$\begin{aligned}
(a_1x_1 + a_2x_2 + \cdots + a_kx_k) \bmod 251 &\equiv b_1 \\
(a_{k+1}x_{k+1} + a_{k+2}x_{k+2} + \cdots + a_{k+k}x_{k+k}) \bmod 251 &\equiv b_2 \\
\vdots &\quad \vdots \\
\left(a_{k(k-1)+1}x_{k(k-1)+1} + \cdots + a_{k(k-1)+k}x_{k(k-1)+k}\right) \bmod 251 &\equiv b_k.
\end{aligned} \tag{3.8}$$

Reliability of this algorithm can be illustrated by using the numerical example given in previous section. Since the example is a $(3, 5)$ scheme, any three of the five shares can be selected during the decoding process. The following coefficient matrix A and right hand side

column vector B are formed if second, third, and fourth shares are selected out of the possible five shares.

$$\left(\begin{bmatrix} 10 & 12 & 2 \\ 18 & 12 & 18 \\ 24 & 27 & 27 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 26 \\ 50 \\ 212 \end{bmatrix} \right) \text{mod} 251. \tag{3.9}$$

Matrix inversion is used to solve this system of linear congruence. The intermediate steps of solution and results are given below.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 & 12 & 2 \\ 18 & 12 & 18 \\ 24 & 27 & 27 \end{bmatrix}^{-1} \begin{bmatrix} 26 \\ 50 \\ 212 \end{bmatrix} = \begin{bmatrix} 24708 \\ 20606 \\ 58806 \end{bmatrix} \text{ mod } 251 \equiv \begin{bmatrix} 110 \\ 24 \\ 72 \end{bmatrix}. \tag{3.10}$$

All pixels of three shares in the form of three subpixel groups should be processed as shown above to fully reconstruct the secret image.

4. Experimental Results

The proposed algorithm is tested to show the feasibility of the scheme. First experiment is selected to be a $(3, 5)$ threshold case to prove the correctness of the method. Gray-level test images from the USC-SIPI image database are used for the experiments. Figure 3(a) is the secret image of size 256×256 pixels. Each cover image is also the same size with the secret image. For a $(3, 5)$ scheme, five different gray-level cover images, Lake, Lena, Pepper, Jet, and Baboon are used. Figure 3(b) presents stego images generated by the proposed method with their PSNR values to indicate the relative modifications compared to the originals.

Secret and stego images are of the same size as expected. In other words, secret and cover images used in the distribution algorithm are of the same size as opposed to four times larger cover images used in the previous publications based on the polynomial approach [9–11]. Storage space and transmission time of stego images are important issues in secret sharing. These issues are addressed since the proposed method uses cover images with the same size as the secret image. The cover-to-secret file size ratios of the Lin-Tasi's scheme, Yang's scheme and the proposed scheme, are listed in Table 1. The proposed method uses the geometric approach, whereas other methods listed in the table are based on the polynomial approach and require four times larger cover images compared to size of the secret image. Larger shares need four times the storage or network bandwidth compared to the proposed method.

The second experiment is performed to find out the quantitative quality of the stego images with respect to other methods. Four different threshold schemes are used for this test. While the value of n is set to 5, threshold values of k in the range of [2–5] are used for this experiment. PSNR values of five sets of five stego images remain constant for other methods as listed in Table 2. While the proposed method yields worse results than other methods in a $(2, 5)$ scheme, it exhibits better PSNR performance for threshold values larger than two. PSNR

Table 1: Comparison of cover/secret file size ratios.

	Lin-Tsai	Yang et al.	Proposed Method
Cover Size/Secret Size	4	4	1

(a)

(b)

Figure 3: (a) Secret image (b) Generated stego images.

values of Lin-Tsai's and Yang's stego images remain constant for schemes with different k values, whereas our method yields better results for $k > 2$ independent of the value of n. This experiment proves that the proposed method produces stego images with better visual quality (less modification) than other methods for (k, n) schemes for $k > 3$. Our scheme is also applicable for true color images like other methods by changing the value of modulo operator's operand.

Last experiment is performed to compare the proposed method with other methods based on geometric approach reported in the literature [12, 13]. Though based on the same geometric secret sharing scheme, these methods produce noise-like shares which may draw attention of malicious attackers. Friendly (natural looking) share generation has become an important issue, and the proposed method produces friendly shares by using Steganography different from the other two methods that are based on geometric approach.

Table 2: PSNR's of stego images for other two schemes and proposed scheme.

(2, 5) scheme	Lake	Lena	Pepper	Jet	Baboon
Lin-Tsai	38.49	38.6	38.29	38.35	37.71
Yang et al.	40.97	41.1	40.66	40.15	40.06
The Proposed Scheme	33.4	33.37	33.85	34.38	33.69
(3, 5) scheme	Lake	Lena	Pepper	Jet	Baboon
Lin-Tsai	38.49	38.6	38.29	38.35	37.71
Yang et al.	40.97	41.1	40.66	40.15	40.06
The Proposed Scheme	38.74	38.45	38.32	38.41	38.17
(4, 5) scheme	Lake	Lena	Pepper	Jet	Baboon
Lin-Tsai	38.49	38.6	38.29	38.35	37.71
Yang et al.	40.97	41.1	40.66	40.15	40.06
The Proposed Scheme	43.58	43.3	43.21	43.16	42.95
(5, 5) scheme	Lake	Lena	Pepper	Jet	Baboon
Lin-Tsai	38.49	38.6	38.29	38.35	37.71
Yang et al.	40.97	41.1	40.66	40.15	40.06
The Proposed Scheme	45.45	45.07	44.88	44.95	44.65

5. Conclusion

This paper proposes a method that uses Blakley's secret sharing concept with Steganography and authentication. There are some reports in literature that use Steganography with Shamir's secret sharing method. However, cover images used to create shares must be four times larger than secret image. For a secret image of $N \times N$ pixels, cover images should be of size $2N \times 2N$ pixels. Such a restriction consumes bandwidth during transmission of the cover images and requires more storage capacity. The proposed method uses $N \times N$ pixel cover images to share an $N \times N$ secret image improving the storage and bandwidth requirements of the friendly secret image sharing techniques.

Therefore, using geometric approach in secret sharing enhanced with Steganography improves two important criteria: storage capacity and bandwidth requirement. Moreover, published works in the literature that use geometric approach produce noise-like shares, whereas the proposed method produces meaningful shares. Meaningful share generation is another important criterion to evaluate the success of a secret sharing scheme because noise-like shares draw attention of the malicious attackers.

Experimental results show that the proposed method uses cover images smaller than other secret image sharing methods based on the polynomial approach reported in the literature. Steganographic image quality of the shares created by this method is better and yields higher PSNR values compared to methods with polynomial approach for k values greater than two.

References

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[2] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the National Computer Conference*, pp. 313–317, New York, NY, USA, June 1979.
[3] C.-C. Thien and J.-C. Lin, "Secret image sharing," *Computers and Graphics*, vol. 26, no. 5, pp. 765–770, 2002.

[4] R.-Z. Wang and C.-H. Su, "Secret image sharing with smaller shadow images," *Pattern Recognition Letters*, vol. 27, no. 6, pp. 551–555, 2006.
[5] C.-C. Chang, C.-C. Lin, C.-H. Lin, and Y.-H. Chen, "A novel secret image sharing scheme in color images using small shadow images," *Information Sciences*, vol. 178, no. 11, pp. 2433–2447, 2008.
[6] R. Zhao, J.-J. Zhao, F. Dai, and F.-Q. Zhao, "A new image secret sharing scheme to identify cheaters," *Computer Standards and Interfaces*, vol. 31, no. 1, pp. 252–257, 2009.
[7] P.-Y. Lin, J.-S. Lee, and C.-C. Chang, "Distortion-free secret image sharing mechanism using modulus operator," *Pattern Recognition*, vol. 42, no. 5, pp. 886–895, 2009.
[8] 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, 2003.
[9] C.-C. Lin and W.-H. Tsai, "Secret image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 73, no. 3, pp. 405–414, 2004.
[10] C.-N. Yang, T.-S. Chen, K. H. Yu, and C.-C. Wang, "Improvements of image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 80, no. 7, pp. 1070–1076, 2007.
[11] C.-C. Chang, Y.-P. Hsieh, and C.-H. Lin, "Sharing secrets in stego images with authentication," *Pattern Recognition*, vol. 41, no. 10, pp. 3130–3137, 2008.
[12] C.-C. Chen and W.-Y. Fu, "A geometry-based secret image sharing approach," *Journal of Information Science and Engineering*, vol. 24, no. 5, pp. 1567–1577, 2008.
[13] H.-K. Tso, "Sharing secret images using Blakley's concept," *Optical Engineering*, vol. 47, no. 7, Article ID 077001, 3 pages, 2008.