Random Matrices Based Image Secret Sharing

Abstract: This paper presents an image secret sharing method based on some random matrices that acts as a key for secret sharing. The technique allows a secret image to be divided into four image shares with each share individually looks meaningless. To reconstruct the secret image all four shares have to be used. Any subset m (1 ≤ m ≤ 4) shares cannot get sufficient information to reveal the secret image. The share generation algorithm works by converting three pixels of the secret image to one pixel each of four different shares based on four random matrices. So, each share is reduced by 1/3rd of the original secret image. During reconstruction, one pixel of each shares are used to reconstruct three pixels of original secret image using the same set of random matrices by image reconstruction algorithm. Our method of share generation is an effective, reliable, and secure method to prevent the secret image. The advantages of this approach in comparison with other image secret sharing methods are its large compression rate on the size of the image shares, its strong protection of the secret image and its ability for real time processing.

Keywords: Secret Sharing; threshold cryptography; shadow images; security

I. INTRODUCTION

Secret images are used in many commercial and military applications. The prime concerns in these applications are the storage and transmission security of certain secret images. To increase the security of secret images, many techniques like traditional encryption, image hiding [16], watermarking [17], steganography [15] etc. are proposed in recent years. A common weakness of the entire above-mentioned security techniques viz. image hiding, watermarking and steganography is that the secret image is stored and transmitted as a single unit. If that single unit is somehow captured by an intruder, the secret may not remain secret. Secret sharing method on the other hand divides a secret into some components called shadow images where each shadow image looks meaningless. Secret image sharing [1] is the art and science about the protection of important images by distributed storages. The concept of secret sharing was proposed by Blakley [1] and Shamir [5] independently in 1979. Secret sharing refers to the method of distributing a secret media like image amongst a group of participants. Each participant is allocated a share of the secret that looks meaningless. The secret can be reconstructed only when a sufficient number of shares are combined together. The sharing is performed in such a way that only certain specified subsets of players are able to reconstruct the secret, while smaller subsets have no information about this secret at all. More formally, in a secret sharing scheme there are one-dealer and n players. The dealer accomplishes this by giving each player a share in such a way that any group of t (for threshold) or more players can together reconstruct the secret but no group of fewer than t players can reconstruct the secret. Such a system is called a (t, n)-threshold scheme.

Shamir [5] developed the idea of a (k, n) threshold based secret sharing technique (k ≤ n). The technique allows a polynomial function of order (k - 1) constructed as,

\[ f(x) = (d_0 + d_1 x + d_2 x^2 + \ldots + d_{k-1} x^{k-1}) \mod p \]

where the value \( d_0 \) is the secret and p is a prime number. The secret shares are the pairs of values \((x_i, y_i)\) where \( y_i = f(x_i) \), \( 1 \leq i \leq n \) and \( 0 < x_1 < x_2 \ldots < x_n \leq p - 1 \).

The polynomial function \( f(x) \) is destroyed after each shareholder possesses a pair of values \((x_i, y_i)\) so that no single shareholder knows the secret value \( d_0 \). In fact, no groups of \( k - 1 \) or fewer secret shares can discover the secret \( d_0 \). On the other hand, when k or more secret shares are available, then one can set at least k linear equations \( y_i = f(x_i) \) for the unknown \( d_i \)'s. The unique solution to these equations shows that the secret value \( d_0 \) can be easily obtained by using Lagrange interpolation.

In 2002, Thien and Lin [6] proposed a (k, n) threshold based secret image sharing scheme by cleverly using Shamir’s secret sharing scheme [5] to generate image shares. The essential idea is to use a polynomial function of order (k - 1) to construct n image shares from an \( l \times 1 \) pixels secret image (denoted as I) as,

\[ S_s(i, j) = (I(ik + 1, j) + I(ik + 2, j)x \ldots + I(ik + k, j)x^{k-1}) \mod p \]

where \( 0 \leq i \leq \left\lfloor \frac{l}{k} \right\rfloor \) and \( 1 \leq j \leq l \). This method reduces the size of image shares to become 1/k of the size of the secret image. Any k image shares are able to reconstruct every pixel value in the secret image. Thien and Lin also...
provided some research insights for lossless image recovery using their technique. They further introduced the possibility of a steganography approach [6, 13] by hiding image shares into host images.

Bai [14] developed a secret sharing scheme using matrix projection. The idea is based upon the invariance property of matrix projection. This scheme can also be used to share multiple secrets.

Wang and Su [11] proposed a secret image sharing method using Huffman coding. In 2008, Shi et al. [12], proposed a new scheme for image encryption based on Shamir’s secret sharing, where the size of each share is \(2(\log_2 m)^2/m^2\) of that of the shared \(m \times m\) image. Their reconstructed matrix is the same as the secret matrix and the shares are \(1/m\) of the size of the secret matrix. Its main advantages are multiple secrets sharing, strong protection of the secrets and smaller size for the secret shares.

In this paper, we propose a secret image sharing method where all shares are needed to get back the original image. Our method generates shadow images that are smaller than that of the secret image. The rest of this paper is organized as follows. Section 2 introduces our secret sharing method. The experimental result is shown in Section 3. In Section 4, we provide the security analysis and the benefits of the size reduction of the shadow images. Finally, the conclusions are stated in Section 5.

II. TYPE OUR SHARING ALGORITHM

In this section, we propose our sharing algorithm based on random matrices. We are dividing a secret image into four shares by using random matrix look up procedure. We generate four random matrices named \(R_1, R_2, R_3\) and \(R_4\). The consecutive three pixel values of the 8-bit secret image are grouped to form a 24 bit string. For example, if the first three pixel values of a secret image are 160, 161 and 161. Converting the pixel values into binary form and placing them together will give the bit pattern shown in Fig 1.

![Figure 1. Bit patterns](image)

These 24 bits are then divided into 8 groups of 3 bits each as shown in Fig 2.

![Figure 2. Grouping of 3 pixels in 8 group](image)

Then we consider \(g_0\) and \(g_2\) as row and column indices to find a value form first random matrix \(R_1\) and put that value as the first pixel of share 1. Say there is 168 in \(R_1\) at location \(g_0, 101\) and \(g_2, 001\). This value 168 is put into first shadow image \(S_1\). Similarly, with other groups we look up at the other random matrices and create other shadow images. The detail process is outlined in the algorithm 1.

While reconstructing, we read a value form first shadow image \(S_1\) and find that value in \(R_1\). The location of that value will give \(g_0\) and \(g_2\). Similarly using shadow images \(S_2, S_3\) and \(S_4\) and random matrices \(R_2, R_3\) and \(R_4\), we can get other \(g_i\) values. The complete reconstruction algorithm is given in algorithm 2.

Algorithm 1: Share Generation

**Input:** A gray level secret image \(S\) of size \(M \times N\) and four random matrices of size \(8 \times 8\) denoted by \(R_1, R_2, R_3\) and \(R_4\).

**Output:** Four shadow images of size \(M \times N/3\) denoted by \(S_1, S_2, S_3\) and \(S_4\).

**Steps**

A. Decompose \(S\) into \(M \times N/3\) number of blocks of size 3-pixels in row major orders.

B. For each 3 pixels block
   a) Obtain 24 bits from 3 eight-bit pixels
   b) Divide the 24-bit pixel values into 8 groups \(g_0\) to \(g_7\) of 3 bits each.
   c) Find a value from \(R_1\) using \(g_0\) and \(g_2\) as row and column indices and put that value in \(S_1\).
   d) Find a value from \(R_2\) using \(g_1\) and \(g_3\) as row and column indices and put that value in \(S_2\).
   e) Find a value from \(R_3\) using \(g_4\) and \(g_6\) as row and column indices and put that value in \(S_3\).
   f) Find a value from \(R_4\) using \(g_5\) and \(g_7\) as row and column indices and put that value in \(S_4\).

C. End.

![Figure 3. The share generation technique](image)

![Figure 4. The reconstruction technique](image)
Algorithm 2: Reconstruction from shares

Input: Four shadow images $S_1, S_2, S_3$ and $S_4$ of size $M \times N/3$ and four random matrices of size $8 \times 8$ denoted by $R_1, R_2, R_3$ and $R_4$.

Output: Reconstructed secret image $S'$ of size $M \times N$.

Steps
A. Make 8 groups $g'_0, g'_1, g'_2, g'_3, g'_4, g'_5, g'_6, g'_7$ of 3 bit each.
B. For each pixel of each shadow image
   a. Find the location of a pixel of shadow image $S_1$ in $R_1$. The row and column index of $R_1$ gives $g'_0$ and $g'_2$.
   b. Find the location of a pixel of shadow image $S_2$ in $R_2$. The row and column index of $R_2$ gives $g'_1$ and $g'_3$.
   c. Find the location of a pixel of shadow image $S_3$ in $R_3$. The row and column index of $R_3$ gives $g'_4$ and $g'_5$.
   d. Find the location of a pixel of shadow image $S_4$ in $R_4$. The row and column index of $R_4$ gives $g'_6$ and $g'_7$.
   e. Make a 24-bit sequence $G$ from $g'_0, g'_1, g'_2, g'_3, g'_4, g'_5, g'_6, g'_7$.
   f. Divide $G$ into 3 parts $G_0, G_1, G_2$ of 8-bit each and place them in $S'$ to get 3 pixels of reconstructed image.
C. End.

III. COMPLEXITY ANALYSIS

The computational complexity of our share generation algorithm is $O(MN)$ where $M$ and $N$ are the width and height of the secret image. For each three-pixels of secret image, the share generation algorithm generates a pixel in all four different shares using four random matrices. So, it has to traverse the whole image matrices which give the time complexity of $O(MN)$ where $M$ and $N$ are the width and height of the secret image. During reconstruction, for each pixel of a shadow image it searches that entry in a random matrix of size $16 \times 16$ and reconstructs a subpart of 3-pixel group of original image. To get the complete 3-pixel group of original image, it has to search four random matrices of size $16 \times 16$. These operations give a time complexity of $(16 \times 16 \times 4 \times M \times N/3)$ where $M$ and $N$ are the width and height of the secret image. Using asymptotic notations, the complexity is $O(MN)$ where $M$ and $N$ are the width and height of the secret image. The reconstruction is bit slower than share generation.

Table I. PSNR Values for Share Construction Using our Algorithm

<table>
<thead>
<tr>
<th>Image Name</th>
<th>PSNR Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena.jpg</td>
<td>31.55dB</td>
</tr>
<tr>
<td>Lady.jpg</td>
<td>30.85dB</td>
</tr>
<tr>
<td>Child.jpg</td>
<td>31.17dB</td>
</tr>
<tr>
<td>Fly.jpg</td>
<td>31.25dB</td>
</tr>
<tr>
<td>Cameraman.jpg</td>
<td>31.43dB</td>
</tr>
<tr>
<td>Duck.jpg</td>
<td>30.57dB</td>
</tr>
<tr>
<td>Airplane.jpg</td>
<td>30.42dB</td>
</tr>
</tbody>
</table>

Our method reduces the size of each shadow image to 1/3 of the secret image. The small size of each shadow image is a good property in practice. Besides the saving of storage space or transmission time, some other benefits also exist. For example, we can easily use data hiding technique to hide each shadow image in some other images called host images so that an intruder cannot notice the existence of the shadow image.

IV. EXPERIMENTAL RESULTS

We have done our experimentation in Matlab running on Microsoft Windows XP system with a Pentium® Dual Core Processor having 2 GB RAM. For our experiment, we have used six different images. The names of the images are: Lady.jpg, Child.jpg, Fly.jpg, Cameraman.jpg, Airplane.jpg and duck.jpg. Due to space limitation, we have shown here the results obtained on two images only. Our first secret image is Lena image of size $512 \times 512$, which is shown in Figure 1. The share generated by our algorithm using 4 random matrices on Lena image of Figure 1 is shown in Figures 2 to 5. The shares are of size $512 \times 170$. The reconstructed image obtained by the four shares images of Figure 2 to 5 and random matrices is shown in Figure 6. The reconstructed image is not lossless but the loss is tolerable as the secret is readable. To measure this loss, we have used peak signal to noise ratio (PSNR) metric.

![Figure 5. The secret Image: Lena](image)

![Figure 6. First Share](image)

The Peak Signal to Noise Ratio (PSNR) is defined as

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \ dB$$

where MSE is the mean-square error between the cover image and the stego image. If the cover image is sized $r \times c$, MSE is defined as
\[
MSE = \frac{1}{r \times c} \sum_{i=1}^{r} \sum_{j=1}^{c} (x_{ij} - y_{ij})^2
\]

where \(x_{ij}\) and \(y_{ij}\) denote the cover and the stego pixel values, respectively.

The PSNR values calculated on various secret images and their respectively reconstructed images are given in Table 1. The PSNR values are greater than 30 in all cases, which is acceptable in several applications of secret image sharing. The loss is due to repeated entry of the same value in random matrices. If all the values of the random matrices are unique then the reconstructed image is lossless. Since, the matrices are generated randomly; we cannot assure all unique values in those random matrices.

V. CONCLUSION

We proposed a method such that secret image can be divided into four shadow images each of which alone is meaningless. The size of each shadow image is 1/3 of the secret images. Due to the small size property, our method gets certain benefits like easier process for storage, transmission, and hiding. The proposed method does not need complicated computation to generate shares and reconstruct original image but it needs all the shares to get back the original image. The probability of reconstruction of the message from individual shares is very less so this method ensures satisfactory results in the field of security. The reconstructed image is lossy because of repeated entries in random matrices. The authors are currently looking for better ways to generate matrices without repeated entries. The other limitation of our method is that it is slow during reconstruction. For reconstruction of every pixel, all the four random matrices have to be searched for the specific value to get the desired pixel of the original image. The searching time is constant with a value of \((4 \times 16 \times 16)\) but is a great overhead in moderate size images. The authors are engaged in finding ways to reduce this searching overhead to reduce the overall time needed to reconstruct the original image.

VI. ACKNOWLEDGMENT

The authors acknowledge the support given by the management and staff of Academy of Technology while carrying out this work.

VII. REFERENCES


