An Investigation on Image Secret Sharing

Saman Salehi \(^1\) and Mohammad Ali Balafar

Faculty of electrical and computer science, University of Tabriz, Iran

Abstract

Numerous multimedia applications and communications are rapidly growing through the Internet. Because most of these multimedia communications are confidential and cannot be known by unauthorized users, secret image sharing has become a key technology for digital images in secured storage and confidential transmission. The ultimate objectives of secret sharing include transparency, authority (resistance to various image tampering and forgery methods), and high capacity of the hidden data. A variety of secret image sharing algorithms and techniques have been proposed in order to satisfy these objectives. These goals were used in this paper to compare current methods of image secret sharing and present their properties and limitations. This paper provided a state-of-the-art review and analysis of different existing secret image sharing methods along with some common standards and guidelines drawn from the literature.

Keywords: Secret Sharing, Steganography, Data Hiding

1. Introduction

Along with the quick development of communication and networking, copying, publishing and transmitting digital multimedia via the Internet is quite convenient. How effective processing and managing of sensitive information have become an important topic to be considered nowadays. Sensitive data can be managed by cryptography or steganography technique. Cryptography is a study to find technique for secure communication in presence of third parties. Steganography is the art of communicating hidden information in such a way that no one doubts the existence of the message. The Steganography is a Greek word that means protected writing. The advantage of steganography over cryptography is that secure information does not attract attention to itself. In cryptography, security depends on the length of the secret keys. Normally, cryptography is used in digital communications, computer networks, and computer security. Also, steganography is usually applied in various types of digital media such as image, video, and audio nowadays. Secret image sharing is one of the common steganography methods, which is a mechanism to protect a secret image among a group of participants by encrypting the secret into shares and decrypting the secret with sufficient shares. Secret sharing scheme can be used for situations where permission to access the critical information depends on several people, not on an individual. This scheme share secret data into several pieces and independently keeps a secret share. So, without all \( k \) participants, critical or secret information cannot be retrieve from individual share. The goal of this paper is to use an image as a carrier and secret sharing as on information hiding technique.

\(^1\)Faculty of electrical and computer science, University of Tabriz, Iran Tel.: +98-935-4165439 tacsaman@gmail.com
Secret sharing can be used in many real-live applications. For example, imagine a company with a secure information and common account balance. To access this account balance or secure information, mutual agreements of all the participants are needed. Another example is bank safe. Secret sharing can be used for accessing the bank safe. The secret password can be shared among several images. Then, these shares are given to participants ((k; n) scheme) and MMS (Multimedia Message Service) application service is implemented for retrieving secret password and accessing to the bank safe.

The rest of this paper contains four sections. In Section 2, visual secret sharing scheme (VSS) is described. In Section 3, secret sharing scheme in the spatial domain is considered. In Section 4, secret sharing is considered in frequency domain. Section 5 is devoted to the analysis and recommendations.

2. Visual Secret Sharing (VSS)

Visual secret sharing is a technique for encrypting secret images into several noise-like share images and then decoding the secret by stacking these shares. In 1995, N. Noar (1995) proposed the idea of visual cryptography (VC) for the first time. This idea provides a fast and easy decryption process for encoding a binary image into transparent and opaque parts. For revealing a binary secret image, shares are stacked together. The advantage of this method is that for revealing secret image, no computation and computer hardware are required. It can be directly decoded by human visual system (stacking two shares). However, this method can be only used for binary images, and shares are noise-like. So, noise-like shares may be suspicious to invaders. Several researchers have used visual cryptography for color and gray-level images such as (Verheul and van Tilborg, 1997; Blundo, et al., 2000; Lin and Tsai, 2003).

Visual cryptography has three main drawbacks: expanding shares size, needing for the codebook to be redesigned for specific applications, and limitation in encoding only one secret image at a time. For overcoming the first two drawbacks, Kafri and Keren (1987) proposed a VSS method known as random grid (RG) (Chen and Tsao, 2008; Shyu, 2007, 2009). Using RG method, a binary secret image is shared into two noise-like cipher-grids that are the same size as the original secret image. In Figure 1, an example of secret sharing based on RG is presented. Several researchers have proposed novel methods for overcoming the third drawback. Chen and Li (2012) presented a new scheme that combined the techniques of circular VC (Wu and Chang, 2005) and random grids to generate circular cipher-grids. As a result, they achieved the goal of sharing

T-Her Chen, *et al.*, combined circular image sharing and RG to propose a novel algorithm that needed no codebook, no pixel expansion and can encode more than one secret image in two shares. For creating the first share ($G_1$), a circular cipher grid with random sectors was created which were transparent or opaque. The number of transparent sectors and opaque sectors was equivalent. The process of creating the second cipher grid are shown below and its complete diagram is illustrated in Figure 2. The following demonstrates the steps of this algorithm.

Step 1: Treat the binary secret image as a one-dimensional array of messages 0 and 1.
Step 2: Compare a binary secret message 0 or 1 with the corresponding sector value of cipher grid A ($G_1$) individually to generate a sector-pixel for cipher grid B ($G_2$) by Algorithm RGVSS.

Although, there are several different algorithms for implementing RGVSS, one of them is introduced here. The RGVSS is shown in Algorithm 1. The interested reader can refer to Shyu (2009) for other algorithms. In the decryption process, for revealing the first secret image, two shares are stacked. For revealing other secret images, one of the stacked shares is rotated in a circular clockwise mode in a fixed degree.

---

**Algorithm 1 Random Grid Visual Secret Sharing (RGVSS)**

Data: A binary secret image $SI=\{(i,j) \mid SI(i,j)=0 \text{ or } 1, 0 \leq i \leq w, 0 \leq j \leq h\}$

Result: Two cipher-grid $G_i=\{(i,j) \mid G_i(i,j)=0 \text{ or } 1, 0 \leq i \leq w, 0 \leq j \leq h\}$

for $i$ and $j$, $0 \leq i \leq w$, $0 \leq j \leq h$ do
  $G_1=\text{random}(0,1)$ //randomly generate "0" or "1"
end

for $i$ and $j$, $0 \leq i \leq w$, $0 \leq j \leq h$ do

---

![Diagram](image_url)
if SI(i,j)==0 then
    \[ G_2(i,j) = G_1(i,j) \]
else
    \[ G_2(i,j) = G_1(i; j); \]
end
returnOutput(G_1,G_2)

Some researchers have proposed methods for improving the visual quality of revealed secret image. Chen and Wu (2011) used a multi-secret image sharing based on boolean operations. Their method was confidential and useful for net-work communications. The secret image can be binary, gray-level, or color and reconstruction of a secret image in this method was distortion-less. In the following, we describe this algorithm brie y. In the first step of image sharing algorithm a random matrix is generated \((S_0)\) by boolean operations. In the second step, then 1 random matrix is generated by \((B_i = SI_i \oplus S_{\oplus 1}, (1 \leq i \leq n - 1))\) equation. Then, share images are created by following operation:

\[
S_i = \begin{cases} 
B_i & \text{if } i = 1 \\
B_i \oplus B_{i-1} & \text{if } i=2,\ldots,n-1 \\
SI_0 \oplus B_{i-1} & \text{if } i=n
\end{cases} \tag{1}
\]

Where SI is secret image. In the decoding phase, to reconstruct the rst secret image all shares are utilized by this equation: \(SI_1 = S_1 \oplus S_2,\ldots,S_n\)

Then, \(n - 1\) random matrices are reconstructed by:

\[
B_i = \begin{cases} 
S_i & \text{if } i = 1 \\
S_i \oplus S_{i-1} & \text{if } i=2,\ldots,n-1
\end{cases} \tag{2}
\]

Finally, other \(n - 1\) secret images are revealed by the following operation:

\[
SI_k = B_i \oplus S_k \text{ for } k = 1,\ldots,n-1 \tag{3}
\]

Encoding and decoding phases of this algorithm are drawn in Figure 3. The most important disadvantage of boolean operators is that shares are noise-like. So, it attracts attacker attention. The advantage of this method is that reconstructed secret image, in a loss-less format, has no pixel expansion and no coobock is required.

Visual quality and security are two important factors for analyzing VSS schemes. The visual quality represents how much the stacked result of two cipher-grids discloses the secret image to the naked human eyes and security means how di cult the information in a secret image is disclosed by one share individually. Visual quality is qualitative while security is quantitative. For more information about visual quality and security, the interested reader can refer to Chen and Li (2012). A summary of various VSS schemes are presented in Table 1 and the functionality of these schemes is compared in important aspects.

VSS based on visual cryptography di ers in two aspects from the image sharing by steganography. First, VSS can be directly decoded by the human visual system and need no computer. Second, VSS is desirable only for binary images. To encrypt information in gray-scale or color images, image sharing by steganography must be used instead of visual cryptography. Steganography based secret image sharing methods
are categorized in spatial and frequency domains. In the next section, secret image sharing in the spatial domain is introduced.

![Figure 3. Multi Secret Image Sharing by Boolean-based Visual Secret Sharing, the Left Figure shows the encoding phase, the Right Image Shows Decoding Phase](image)

**Table 1. Compare of Several VSS Methods in Multiple Important Aspects**

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Encryption method</th>
<th>Decoded needed</th>
<th>Share size in computer</th>
<th>Multiple secret</th>
<th>Secret share</th>
</tr>
</thead>
<tbody>
<tr>
<td>N. Kaur (1995)</td>
<td>VC</td>
<td>Yes</td>
<td>500x500</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Wu and Cheng (2000)</td>
<td>VC</td>
<td>Yes</td>
<td>400x400</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Shi et al. (2002)</td>
<td>VC</td>
<td>Yes</td>
<td>400x400</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Pan (2000)</td>
<td>VC</td>
<td>Yes</td>
<td>500x500</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Liu and Xia (2000)</td>
<td>EG</td>
<td>No</td>
<td>300x300</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Shi et al. (2002)</td>
<td>EG</td>
<td>No</td>
<td>300x300</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>T.H. Chan (2000)</td>
<td>EG</td>
<td>No</td>
<td>400x400</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Chan and Tso (2000)</td>
<td>EG</td>
<td>No</td>
<td>200x200</td>
<td>Yes (Gray)</td>
<td>Yes</td>
</tr>
<tr>
<td>Chan and Li (2002)</td>
<td>EG</td>
<td>No</td>
<td>256x256</td>
<td>Yes (gray)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 3. Secret Image Sharing in Spatial Domain

In 1979, C. Shamir (1979) proposed the theory of secret sharing scheme based on Lagrange’s polynomial interpolation. He showed how to divide data into pieces in such a way that it is easily reconstructable from any k pieces, but even complete knowledge of k-1 pieces reveals absolutely no information about D. This scheme is known as (k; n) threshold secret image sharing. Shamir introduces his method with this conception: "Ideally we would like the cooperation to be based on mutual consent, but the veto power this mechanism gives to each member can paralyze the activities of the group. By properly choosing the k and n parameters we can give any sufficiently large majority the authority to take some action while giving any sufficiently large minority the power to block it."

In 2002, Thien and Lin (2002) proposed a (k, n) method to share a secret image into shadow images which were noise-like. To restore the whole image, any k shadow images (k ≤ n) could be used. In Thien, et al., method, size of each share is smaller than the secret image. Thus, this method is desirable for situation in which storage and transmission are important factors. Thien et al.’s method has steps for creating each
shares. In the rst step, all gray values larger than 250 are converted into 250. So, the all gray levels fall in the range of 0 to 250. In the second step, for increasing security, the secret image is scrambled by a permutation key. In the third step, k not-shared-yet pixels of the permuted image are sequentially taken \((a_{a_1} \ldots a_{a_{k-1}})\). In the fourth step, every section generated in step three is given as the input of equation 4 which generates pixel values \(q(x_i) \ldots q(x_n)\). These pixels are used for generate shadow images. Finally, steps three and four are repeated.

\[ q_j = a_0 + a_1 x^1 + \ldots + a_{j-1} x^{j-1} \mod 251 \tag{4} \]

In order to make interpolation to be possible, the prim number which is used in module operation, must be bigger than a and \(k\) \((a \leq 250, k \leq 250)\). That is why 251 were chosen in equation 4. In Figure 4, experimental results of Thein, et al., procedure can be observed. The size of every share in this method is 1=k of the secret image. The disadvantage of this method is that share images are noise-like and may attract attacker’s attention. To overcome this disadvantage, Thien and Lin (2003) proposed a user-friendly image sharing method for easier management of the shadow images. In this method, they use natural images in place of noise-like images (Figure 5) for sharing procedure. In 2004, Lin and Tsai (2004) proposed novel image secret sharing method based on threshold scheme which has additional capability such as authentication. The authentication in this method was done by parity-bit checking with embedding fragile watermark signals into the cover images. A fragile watermark (Lin and Delp, 1999) is a kind of signal which is designed to be embedded in an image and can be easily destroyed if the watermarked image is manipulated in the slightest manner. Let’s explain Lin, et al., method using an example:

Assume \(S\) as a single pixel of an image taken as the secret image, and \(B_1, B_2, \ldots, B_n\) as \(n\) distinct gray-scale image blocks with size \(2 \times 2\) pixels. \(X_i, W_i, V_i\) and \(U_i\) represent four pixels of each block and their values as \(x, w, v\) and \(u\) respectively. \(b_1, b_2, \ldots, b_n\) are data bits to be used as watermark signals. Note that, \(b_i\) are embedded in \(B_i\) respectively. In the first step, they select the top left most pixel for \(x\) value and corresponding secret pixel for \(y\) and an arbitrary set
Figure 4. The (2, n) Secret Image Sharing (a) A 512x512 Secret Images; (b) The Permuted Image of (a); (c) (f) The Four Shadow Images Whose Size are All 512 *256 Bytes; (g) The Image Revealed from Any Two of These Four Images Thien and Lin (2002)
Numbers form $1, \ldots, m_{k-1}$ (Equation 5). In the second step, for each $x_i$, the corresponding value of $F(x_i)$ is computed by Equation 5 to form a secret share $(x_i, F(x_i))$. In the next step, 8 bit data of $F(x_i)$ are shared in 2LSB of $W_i$, $V_i$ and $U_i$. Finally, fragile watermark bits are stored in the third LSB bit of $W_i$ (see Figure 6). These steps are repeated for hiding all the secret pixels. In Figure 7, the authenticating example of Lin and Tsai scheme is given for tamper detection.

$$F(X) = y + m_1x + m_2x^2 + \ldots + m_jx^{j-1} \mod 251$$

They also extended their scheme to handle color images. First disadvantage of this algorithm is that, for authentication, all blocks of every shares should be verified.
**Figure 6. Illustration of Cover Block After Embedding Secret Pixel and Authentication Bit (Fragile Watermark), Authentication Bit is Parity Check Bit**

<table>
<thead>
<tr>
<th>$X_i$</th>
<th>$W_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_i = x_{i1}x_{i2}...x_{i8}$</td>
<td>$w_i'' = w_{i1}w_{i2}...w_{i8}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$V_i$</th>
<th>$U_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_i' = v_{i1}v_{i2}...v_{i8}$</td>
<td>$u_i' = u_{i1}u_{i2}...u_{i8}$</td>
</tr>
</tbody>
</table>

**Figure 7. Secret Sharing with Authentication for Tamper Detection, (a) The Stego-image in which Fragile Watermark Signals are Embedded (b) The Image with Modifications Added to (a), (c) The Result of Authentication (Lin and Tsai, 2004)**

By complexity time $8^* n^*m^2$ with order $O(m^2)$. $T_{n^2}$ is the number of pixels of secret image, $n$ is the number of shares or participants and $8$ is the number of bits which must be checked for parity (even or odd). The second disadvantage is that participants can easily find parity check according to $w_{ipixel}$ value. For example, they can change the value of pixel $w_{i}$ from 10101111 to 11011000 which satisfy the same even parity check policy, but restoring of secret image is impossible. Also, because this scheme uses three bits of each pixel (pixel of cover image) for hiding information, image quality of stego-image is deteriorated. To overcome these weaknesses, Yang et al. (2007) proposed an improved scheme with authentication ability that prevents dishonest participants from cheating, by hash function instead of parity checking. Also, using Galois Field $2^{GF(28)}$, they improved their scheme to a loss-less version. In order to construct lossless image sharing, Yang et al used the power of two Galois Field instead of prim Galois Field $GF(251)$. Thus, the improved version of $(k-1)$ degree polynomial was defined as follows:

$$F(X) = S + a_1x^1 + a_2x^2 + ... + a_{k-1}x^{k-1} \text{ mod } GF(2^8)$$

(6)
In Equation 6, \(x\) is formed by combining \(x_{i1}, \ldots, x_{i6}\) and \(v_{i4}, v_{i3}\) from cover image \(S\) is the secret pixel chosen from secret image, respectively. \(a_1, \ldots, a_{k-1}\) are random numbers and mod is modulo operator. In Yang et al.’s scheme, all the variables are between 0 and 255. So, secret image can be recovered without any deterioration. Yang et al. showed that, for hiding nine bits in a four-pixel block, several arrangements of bits can be used. They demonstrated that the worst arrangement of bits was (0-0-8-1) and the best one was (2-3-2-2). So, they chose the arrangement of (2-3-2-2) for their scheme, which is given in Figure 8. For authentication, they used a hash function with secret key \(K_1\), block index \(b_{id}\) and stego-image identification \(I_{id}\). Moreover, they computed \(n\) hash bits \(p_{i1}^{(1)}, p_{i2}^{(2)} \ldots, p_{in}^{(n)}\) for every block \(B_{ij}\).

Yang et al., used one bit in each block for authentication and stored only one secret pixel at a time in each block of cover image. To improve the stego-image quality, Chang et al., (2008) employed all \(k\) secrets as coefficients in \((k-1)\)-degree polynomial \(F(X_i)\). This \((k-1)\) polynomial is shown in Equation 7.

\[
F(X_i) = a_1 + a_2 X^{1} + \ldots + a_{k-1} X^{k-1} \mod 251
\]  

(7)

where \(a_1, a_2, \ldots, a_{k-1}\) are secret integer values chosen from secret image and \(x\) is integer selected from \(X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})\). To prevent malicious participants and enhance authentication ability, they used 4 bits authentication instead of 1 bit authentication. A random bit stream generator with secret key \(k_2\), \(M^{(i)}_{(j)}\) (six prime modula), and block index \((i, j)\) was used to create four authentication bits, \(p_{i4}, p_{i3}, p_{i2}, p_{i1}\) for each block \(B_{ij}\). For improving authentication ability, they used Chinese remainder theorem (CRT) \(^3\). The property of CRT is that it is computationally infeasible to recover the integer which will be embedded in the cover image using different residue sets.

Figure 8. The 2 *2 Block of Cover Image after Embedding Secret Pixels in 2 LSB, \(p_s\) are Authentication Bits

In Figure 9, Lin and Tsai, Yang et al., and Chang et al.,’s Schemes in Authentication of Image Manipulation.
In comparison to Yang, et al., Chang, et al., scheme is more powerful in authenticating cover images. It embeds secret pixels instead of one secret pixel at a time, therefore, visual quality of the stego-images has been enhanced enough. Other benefit of Chang, et al., scheme is that participants cannot transpose the watermarked blocks in the stego-images to create a collage, because placing the watermarked blocks in the wrong position cannot be successfully verified. In other words, this scheme can detect manipulated block with the probability of \( 15 \cdot 2^{-4} \). If \( m \) tampered blocks are supposed to exist, the probability of successfully verifying all these blocks is:

\[
\prod_{1}^{m} 1/16
\]

In Figure 9, schemes of Lin and Tsai, Yang, et al., and Chang, et al., are compared in terms of authentication ability and tampered detection. The greater the number of bits used for authentication, the higher the capability of detecting fake stego-blocks.

There are other ways for enhancing image quality. As an example, Wu, et al., (2011) employed optimal pixel adjustment process to enhance image quality under different payload capacity and various authentication bit conditions. In order to enhance image quality of the stego-image, optimal LSBs method was proposed by Chan and Cheng (2004). For minimizing embedding error \( (p_i - p_{i-1}) \), they modified four block pixels \( X_i, W_i, V_i, \text{and } U_i \). If error is negative \( (-2^z < p_i < -2^{z-1}) \), number \( 2^z \) is added otherwise if error is positive \( (2^{z-1} < p_i < 2^z) \), number \( 2^z \) is subtracted. This modification process is given in Equation 8, where \( z \) is the number of embedded bits, \( p_i \) is pixel before embedding, \( p_i \) is pixel after embedding and \( p_i = p_i - p_{i-1} \).

\[
p_i' = \begin{cases} p_i - 2^z & \text{if } 2^{z-1} < p_i < 2^z \text{ and } p_i \geq 2^z \\ p_i + 2^z & \text{if } -2^z < p_i < -2^{z-1} \text{ and } p_i < 256 - 2^z \\ p_i & \text{otherwise}. \end{cases}
\]

After performing this procedure, the error range is reduced from \( (2^{z-1} < p_i < 2^z) \) to \( (2^{z-1} < p_i < 2^z) \) (image distortion is reduced). For more details, refer to Wu (2007); Chan and Cheng (2004). Also, to improve authentication and prevent dishonest participants from executing malicious modification, Chun Wu et al. used Equation 9 to generate authentication bits. In this equation, \( H_{k3}(\cdot) \) is a function for generating keyed-hash message authentication code (Castiglione et al., 2010) proposed in IETF RFC 2104 and FIPS 198 HMAC (Hash Message Authentication Code). \( B_{id} \in [1, m \times m] \) is block index and \( I_{id} \in [1, n] \) is identification number of stego-image. \( (X_i - x_{i_1}), (W_i - w_{i_1}), (V_i - v_{i_1}), (U_i - u_{i_1}) \) are bits value of four pixels after embedding secret pixel without authentication bit. Symbol ||, denote concatenation and symbol \( \oplus \), denote XOR operation.

\[
h_i = H_{k3}((X_i - x_{i_1}) || (W_i - w_{i_1}) || (V_i - v_{i_1}) || (U_i - u_{i_1}) || B_{id} || I_{id}).
\]

\[
(p_{i_1}, p_{i_2}, p_{i_3}, p_{i_4}) = (h_{i_{12}}, h_{i_{13}}, h_{i_{14}}, h_{i_{15}}) \oplus ... \oplus ((h_{i_4}, h_{i_3}, h_{i_2}, h_{i_1})) \quad (9)
\]

Chun Wu, et al., Thien\& Lin and Chang, et al., use prime number 251 for module Lagrange interpolation, but, Chung Wu, et al., used optimal pixel adjustment.
Therefore, Chung Wu, et al., method has less image distortion. In comparison to other mentioned methods, Chun Wu, et al., used distinct image identification number, e.g., 1, 2, 3, …, n as the input of polynomial instead of $x_1^i$ for each cover image. In order to avoid the same identification number $I_{id}$, they used one way function (Eslami and Ahmadianabadi, 2011). So, in place of using fixed $I_{id}$ which may have less security, they used one way function to compute $x^{(i)}_1$ from $x^{(i-1)}_1$.

This one way function is computed by Equation 10.

$$x^{(i)}_1 = h(x^{(i-1)}_1) \times 197 \mod 251$$

For each unique value of $x^{(i)}_1$, the integer value of $F(x^{(i)}_1)$ is computed by Equation 7 to form secret shares $F^{(1)}_1, F^{(2)}_1, \ldots, F^{(n)}_1$ respectively. Each pair of $(x^{(i)}_1, F(x^{(i)}_1))$ is a secret share of k secrets. After creating secret shares, these secret shares and authentication bits of previous block are embedded in the LSB pixels of cover block $B^{(0)}$. To compute the first block (at the top-left corner), they store $f_0;0;0;0$ for $P_{1-1;4}$, $P_{1-1;3}$, $P_{1-1;2}$, $P_{1-1;1}$ and in computing the last block (at the bottom-right corner) authentication bits of this block is embedded in the first block. Figure 10 shows the image structure block of Chun Wu, et al., scheme in which $F_8, F_7, \ldots, F_1$ is binary format of secret share $F^{(i)}_1$. In Figure 11, effect of OPAP in enhancing image quality and peak signal-to-noise ratio (PSNR) of image can be seen. This figure demonstrates the comparison result of image quality between Chung Wu, et al., scheme and Lin and Tsais, Yang, et al., and Chang, et al., schemes after applying $(2, n)$-threshold with the same payload capacity. This comparison is based on PSNR. The higher the PSNR value the less the image distortion.

In image secret sharing and steganography, PSNR is the important factor for measuring quantitative comparison. The PSNR measures distortion between original secret image and stego-image.

Figure 10. Image Structure Block of Chun Wu, et al., Scheme
The higher the PSNR value the less the image distortion and possibility of attracting attacker's attention. Generally PSNR is expressed at a logarithmic scale in decibels (dB) and calculated by:

$$PSNR_{dB} = 10 \times \log_{10} \left( \frac{2^{10} \times MSE}{255} \right)$$  \hspace{1cm} (11)

Where MSE is mean square error between cover image and stego-image and is calculate by:

$$MSE = \frac{1}{f \times g} \sum_{i=1}^{f} \sum_{j=1}^{g} (x_{ij} - y_{ij})^2.$$  \hspace{1cm} (12)

**Figure 11. Comparision of (a) Lin and Tsai, (b) Yang, et al., (c) Chang, et al., Chia-Chun Wu, et al., Methods with PSNR**

**Figure 12. The Number of Modified Bits in Each Block of Eslami and Ahmadabadi's Scheme**

Where $f \times g$ is image size, $x_{ij}$ is a pixel of stego-image and $y_{ij}$ is a pixel of cover image. Generally, the human visual system cannot distinguish stego image from cover image if PSNR value be greater than 30. Also, for color image (RGB) average peak signal-to-noise ratio known as APSNR is used as a measurement factor. The APSNR is average PSNR of each color matrices. Detection ratio, $DR = \frac{NTPD}{NTP}$, is the other...
factor that generally be used in precision measuring of image sharing authentication ability. The DR is used for tamper detection ratio of stego-image, where NTP is the number of tamper and NTDPD is the number of tamper detected by scheme.

In literatures, almost all recent secret image sharing algorithms are based on fixed block embedding known as polynomial-based schemes. For example, in Chun Wu et al.’s method (Figure 10), the secret image is forced to be embedded in \(x_{i3}, x_{i2}, w_{i3}, w_{i2}, v_{i3}, v_{i2}, u_{i3}, u_{i2}\) for each block of cover image. In order to reduce cover image distortion, variable-size block based on size of secret image can be used instead of fixed-size blocks. This scheme is known as dynamic embedding. Recently Eslami and Ahmadabadi (2011) proposed a method for dynamic embedding and authentication chaining which can be explained as follow: Supposed that there are \(2^{15}\) blocks with the size of 10 bits (8 bits for shares and two bits for authentication) which should be embedded in cover image with size 512*512. According to Equation 13, the BS (block size based on pixels) is equals to \(2^{18}=2^{15}=8\). Since the number of bits which must be embedded is \(|D| = 10\) and the number of pixels needs for embedding these secret bits is \(BS = 8\), the number of blocks required for embedding is \(Nb\). \(Nb\) is defined as equation 14.

\[
BS = \begin{bmatrix} |CI|, I \end{bmatrix}
\]

(13)

\[
Nb = \begin{bmatrix} |D|, BS \end{bmatrix}
\]

(14)

So, \(Nb = 10/8 = 1.25\). It means that, on average, 1:25 bits of secret image could be embedded in every block of cover image. Because 10 bits of secret image should be embedded in 8 pixels, for the first 6 pixels secret bits are embedded in 1 LSB and for 2 last pixels in 2 LSB, i.e., the number of embedded bits in an 8-pixel block will be \((1, 1, 1, 1, 1, 1, 2, 2)\) (Figure 12). The advantage of dynamic embedding is that all the capacity of cover image can be used for hiding data and there are no restrictions about the size of cover image. Table 2 is the result of applying the mentioned algorithms to three images of Lena, Baboon and pepper. Thus, Eslami and Ahmadabadi’s algorithm is performs better in terms of embedding secret image. Another benefit of Eslami and Ahmadabadi’s method is that with an authentication bit, the probability of successful tampering is \(2^{-2a}\), while it is \(2^{-a}\) in the following methods. Yang (2004); Lin and Tsai (2004); Yang, et al., (2007); Liu, et al., (2009); Tuyls, et al., (2005); Tsai, et al., (2007); Feng, et al., (2008). For achieving this goal, Eslami and Ahmadabadi used authentication-changing. It means that tampering one block of secret share affected other blocks.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Lena</th>
<th>Pepper</th>
<th>Baboon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin et al.</td>
<td>62.29</td>
<td>62.27</td>
<td>62.29</td>
</tr>
<tr>
<td>Yang et al.</td>
<td>44.62</td>
<td>44.58</td>
<td>44.57</td>
</tr>
<tr>
<td>Chang et al.</td>
<td>10.82</td>
<td>10.75</td>
<td>10.21</td>
</tr>
<tr>
<td>Chun Wu et al.</td>
<td>13.54</td>
<td>13.56</td>
<td>13.54</td>
</tr>
<tr>
<td>Eslami and Ahmadabadi</td>
<td>16.13</td>
<td>16.12</td>
<td>16.10</td>
</tr>
</tbody>
</table>

### 3.1. Cellular Automata
All the described methods were based on Shamir’s polynomial. In this section, secret image sharing based on cellular automata will be explained. Computation complexity of cellular automata is of order O(n) while computation complexity of Shamir’s polynomial is O(n log n). Revealing secret image in most of Shamir’s polynomial based schemes is lossy, in contrast, it is loss-less cellular automata. According to the definition by R. Alonso-Sanz (2003) memory cellular automata is a discrete dynamical system formed by a finite number of identical objects called cells. State of cells changes in every discrete step of time by deterministic rule called state transition function. Inputs of this function are the state of cells in the previous time step. Supposed that \( a_{i}^{(T-1)} \) denotes the current state of cell \(<i>\) at time \( T\) with radios \( r \); then, current state is updated by:

\[
a_{i}^{(T)} = f(a_{i-1}^{(T)}, \ldots, a_{i}^{(T)}, \ldots, a_{i+r}^{(T)}), \quad 0 \leq i \leq N - 1
\]  

(15)

In Equation 15, \( N \) is the number of states and \( f(\cdot) \) is linear transformation function which determines value of each configuration cell in the next state, during the evaluation of CA. This function is called local transition function of Linear Cellular Automat (LCA) and has the following form:

\[
a_{i}^{(T+1)} = \sum_{j=r}^{N-1} \alpha_{i,j} a_{i+j}^{(T)} (m \text{ mod } 2), \quad 0 \leq i \leq N - 1
\]  

(16)

This function must be reversible in order to be usable in secret sharing. Evolution backward is proved by Tooli and Margolus (1990). Every state of cellular automata has an initial configuration with symbol \( \mathbb{C}^{(0)} \). The configuration at time \( T \), is denoted by \( \mathbb{C}^{(T)} = (a^{(T)}_{0}, \ldots, a^{(T)}_{N-1}) \). So, fork shares, there are \( k \) evaluation times \( (\mathbb{C}^{(T)})_{u_{0}} \ldots u_{k-1} \). Since, there are \( 2r+1 \) neighbor per cell, there exist \( 2^{2r+1} \) LCAs and each of them has different rule numbers. Rule number represent which of the current configuration cells are used for producing the next configuration.

The rule number is defined as follows:

\[
w = \sum_{j=r}^{N-1} \alpha_{i,j} 2^{r+j}, \quad 0 \leq w \leq 2^{2r+1}-1.
\]  

(17)

The above-considered CA is linear cellular automata. In other words, it is memoryless cellular automata, i.e., the update of the current cell is only depends on neighborhood cells at current time. There is another type of cellular automata known as memory cellular automata which are defined by following local transition function:

\[
a_{i}^{(T+1)} = f_{1}(a_{i-1}^{(T)}, \ldots, a_{i}^{(T)}, \ldots, a_{i+r}^{(T)}) + f_{2}(a_{i-1}^{(T-1)}, \ldots, a_{i}^{(T-1)}, \ldots, a_{i+r}^{(T-1)}) + \ldots
\]  

(18)

Benefit of memory cellular automata is that updating the current state of au-tomata depends on several times not only last time, which makes it possible to use memory cellular automata in secret image sharing. To make use of memory cellular automata in secret sharing, Equation 18 must be reversible. A proof for the reversibility of this explanation can be found in Rey, et al. (2005). InEslami, et al., (2010), a particular type of dynamical system memory cellular automata are employed which is based on \((k, n)\) threshold. Secret image is considered as one of the initial conditions of the memory cellular automata. Es-lami, et al., (2010) proposed new method for steganography based on cellular automata. Their method consists of four phase: (1) the Setup phase, (2) the Sharing phase, (3) the Embedding phase, and (4) the Verification and Recovery phase. In the setup phase, cover image (CI) and sequence number (seq.) are assigned to each participant. The sequence number is used to ensure that participants cannot exchange their shares or announce an incorrect identification number. The public and private key (PU_a, PR_a) are generated for authenticating and
prohibiting tampering of stego-image. In the second phase (sharing phase), secret image is divided into t pixel unit (t is the number of participant). From the pixels of each unit, initial configuration $C^{(0)}$, ..., $C^{(t-2)}$ is created and then $C^{(t-1)}$ is formed from concatenation of $C^{(0)}$, ..., $C^{(t-2)}$ by a collision-free hash function $H(C^{(0)} || ..., || C^{(t-1)})$. This is known as double authentication; the advantage is that, with single test tampered stego image can be detected. $C^{(t-1)}$ is created from concatenating $C^{(0)}$, ..., $C^{(t-2)}$. Therefore, if the cheaters succeed in forging the first authentication, i.e., forging signature (sign), their manipulation will be revealed by this technique. In Chang, et al., scheme all blocks of stego-image should be verified for tamper detection. In the next step of the second phase, $C^{(t)}$, ..., $C^{(t+n-1)}$ is computed from initial configuration. Each of these configurations is assigned to a participant. $SH_1=C^{(0)}$, ..., $SH_n=C^{(t+n-1)}$. Finally, for each participant, an authentication code is calculated by obtained authentication code is stored in cover image. $DS_{PRD}$ is a digital signature with the key $PR_D$.

Figure 13. Embedding in Last Two Bits of Each Block

\[
\begin{align*}
\hat{X} & = x_1 x_2 ... x_6 d_3 d_4 \\
\hat{V} & = v_1 v_2 ... v_6 d_5 d_6 \\
\hat{W} & = w_1 w_2 ... w_6 d_1 d_2 \\
\hat{Z} & = z_1 z_2 ... z_6 d_7 d_8
\end{align*}
\]

In the third phase (the embedding phase), obtained data must be embedded in cover image to create stego-image. For this reason, 8 bits of data are embedded in four blocks of cover image. Embedding will be done in the last two bits of every block (Figure 13). So, this scheme has less distortion.

In the final phase (the verification and recovery phase), if authentication checking is failed, the signature is tampered. Then, the cover image is rejected or recovery phase is used to reveal the share. Afterward, for revealing block $j$ of secret image, first initial configuration is created with $\tilde{C}^{(0)} = SH_{a,1}$, $\tilde{C}^{(1)} = SH_{a,2}$, ..., $\tilde{C}^{(t-1)} = SH_{a,t}$ and $\tilde{M}$ is evolved, $t+\alpha-1$ times, to obtain $\tilde{C}^{(t+n-2)}$, ..., $\tilde{C}^{(t+n-2)}$. For more details about this method, refer to Eslami, et al., (2010). The advantage of cellular automata over Shamir's polynomial is that cellular automata has a synchronous update mechanism for each cell. Second, in Shamir's polynomial, every pixel from 251 to 255 is converted to 250 and prime module 251 must be used to construct shares. But cellular automata do not have this drawback. Third, in cellular automata the last two bits of each block are used for embedding. So, it generates less distortion on cover image in comparison to Shamir's polynomial.
Some researchers such as Wu, et al., (2012) have used cellular automata to reverse secret image in loss-less mode. Image Reversibility in steganography is of great significance. For example, in medical diagnosis and law enforcement, it is very important to reverse the stego-media in loss-less mode (original image). Also, in other applications, such as military image system and remote sensing, it is desirable to recover original media with high precision.

Below, Xiaotian Wu, et al., method is briefly explained. This scheme is based on two-dimensional memory cellular automata. Two-dimensional CA is similar to one-dimensional CA, the difference between 1D and 2D is that the cellular space in 2D is constructed by a finite two dimensional array of \( m \times n \) cells. Here, all of the formulas that describe 1D CA (Equations. 15 and 16 and 17 and 18) are rewritten in 2D.

\[
\begin{align*}
\sigma_{i,j}^{(t+1)} &= f \left( a_{i-1,j-1}^{(t)}, a_{i-1,j}^{(t)}, a_{i-1,j+1}^{(t)}, a_{i,j-1}^{(t)}, a_{i,j+1}^{(t)}, a_{i+1,j-1}^{(t)}, a_{i+1,j}^{(t)}, a_{i+1,j+1}^{(t)} \right), \\
\quad 0 \leq i \leq N, 0 \leq j \leq M
\end{align*}
\]  

(20)

Configuration of the two-dimensional CA, at time step \( t \), is illustrated in Figure 14. Local transition function of 2D CA is given by:

\[
\begin{aligned}
a_{i,j}^{(t+1)} &= \sum_{\sigma \in \Sigma_{i,j}^{(t)}} \lambda_{\sigma} a_{\sigma}^{(t)} \pmod{2} \\
\end{aligned}
\]  

(21)

In rule number, there are 9 neighbors for each cell, so, there exist \( 2^9 \) LCAs, each of which is denoted by:

\[
w = \lambda_{i-1,j-1} 2^8 + \lambda_{i-1,j} 2^7 + \lambda_{i-1,j+1} 2^6 + \lambda_{i,j-1} 2^5 + \lambda_{i,j} 2^4 + \lambda_{i,j+1} 2^3 + \\
\lambda_{i+1,j-1} 2^2 + \lambda_{i+1,j} 2^1 + \lambda_{i+1,j+1} 2^0, 0 \leq w \leq 511
\]

The cellular automata defined above are memory-less. Also the memory CA can be used, in which, the update of cell in time \( t+1 \) depends on time \( t \); \( t \geq 1 \).

This is the basic concept of MCA.

\[
a_{i,j}^{(t+1)} = f \left( N_{i,j}^{(t)}, \ldots, a_{i,j}^{(t-p+1)} \right) \pmod{2}
\]

(22)

For creating stego-image in this method, first, every pixel of secret image is converted into \( k_{\text{ary}}(t) \) notation denoted by \( (S_{ij})_k = k_{\text{ary}}(S_{ij}) \). Then \( k_{\text{ary}}(t) \) secret image is divided by:

\[
(SI)_k = SI_1 \times k^{i-1} + SI_2 \times k^{i-2} + \ldots + SI_i
\]

(23)

After generating initial configuration from RS, RM and \( S_{i1}, \ldots, S_{in} \), local transition function is used to generate next configuration from the current configuration. RM is constructed by random number generator. RS is residual of the cover image. Before
Local transition function is used in Equation 25, this initial configuration must be calculated.

\[
RS = CI \mod k, \\
C^0 = RS \oplus RM, \\
C^1 = S1_i, ..., C^l = S1_i.
\]  

(24)

Local transition function for 2D memory cellular automata is created by:

\[
a^{l+1}_{i,j} = (f_{w_i}(N^{l+1}_{i,j}) + ... + f_{w_j}(N^{l+1}_{i,j})) \mod 2
\]

(25)

Finally, after \( l+u+1 \) evaluation of \( \{c^{(0)}; c^{(1)}, ..., c^{(l)}; c^{(l+1)}, ..., c^{(l+u)}\} \), configurations are generated. The last \( l \) configurations \( c^{(l+1)}, ..., c^{(l+u)} \) can be selected as \( l+1 \) shared data. In camouflage phase or embedding, the last \( l \) configuration is embedded in cover image by module \( k \) operation (Equation 26) to form-stego image.

\[
S_i = \left\lfloor \frac{CI}{k} \times k \right\rfloor + C^{l+u-1}, i \in [1, l+1]
\]

(26)

After embedding secret image in cover image and creating stego-image, the 3-uplet \( (I, w_i, S_i) \) is delivered to the \( i \)-th participants. \( I \) is the participant number, \( w_i \) is rule number and \( S_i \) is stego-image for \( i \)-th participant.

Also, 2D cellular automata can be used for color image sharing. Interested reader can refer to Eslami, et al., (2010) for details of this algorithm. In this method the first issue with 2D cellular automata is related to overflow problem. In camouflage phase, when secret image is embedded in cover image, share's value is added to quantize cover pixel. Therefore, the output stego-pixel value may exceed the gray-scale boundary. So, the pixel value of cover image which are less than \( \frac{2^{56}}{kk} \) should be selected for Xiaotian Wu et al.'s scheme. This XiaotianWus scheme is useful, because secret image can be revealed without distortion in the pixels. In comparison with other methods (Shamir's polynomial) with computational cost of \( O(n \log^2 n) \), computational cost of this method is of order \( O(n) \). This method is reversible (loss-less), so this scheme can be used in patient recording and law enforcement. Although, some other researcher such as Al-Qershi and Khoo (2011) proposed embedding method for medical image based on difference expansion, their embedding capacity has been less than Xiaotian Wu, et al., scheme and their method has not been useful for every type of medical image such as: CT, MRI, X-ray3. Difference expansion (Tian, 2003) is one of the most important schemes in reversible data hiding.

4. Secret Image Sharing in Frequency Domain

LSB embedding mechanism is prefect in terms of not deceiving the human visual system, but, it has weak resistance against attacks. So, researchers have decided to apply LSB embedding in frequency domains. The most important frequency domain algorithms used by researcher are DCT (discrete cosine transform), DWT (discrete wavelet transform) and FT (Fourier transform). In 1997, James S. Walker compared Fourier and Wavelet analysis (Walker, 2011) as steganography techniques. In 2007, Chin Chen Chang proposed a reversible image hiding based on DCT (Chang, et al., 2007). The two-dimensional DCT of input image \( F \) and output image \( G \) is calculated by:

\[
G_{ij} = a_i a_j \sum_{n=0}^{M-1} \sum_{m=0}^{N-1} F_{mn} \cos \left( \frac{\pi (2m + 1) i}{2M} \right) \cos \left( \frac{\pi (2n + 1) j}{2N} \right).
\]

(27)

Where \( 0 \leq i \leq M - 1, 0 \leq j \leq N - 1, a_i \) and \( a_j \) is calculated by:
\[ \alpha_i = \begin{cases} \frac{1}{\sqrt{M}}, & i = 0 \\ \frac{1}{\sqrt{2M}}, & 1 \leq i \leq M-1 \end{cases} \]
\[ \beta_i = \begin{cases} \frac{1}{\sqrt{N}}, & i = 0 \\ \frac{1}{\sqrt{2N}}, & 1 \leq i \leq N-1 \end{cases} \]

\[ (28) \]

Where M, N are dimensions of the input image.

DCT is useful in video and image compression. After getting coefficient of each block DCT from Equation 28, this block is quantized using specific quantization table (Qt). Aim of quantization is to reduce the tightness precision produced by DCT. Quantization step is calculated by the following equation:

\[ g(\omega_x, \omega_y) = \left| \frac{f(\omega_x, \omega_y)}{K(\omega_x, \omega_y)} + \frac{1}{2} \right|, \omega_x, \omega_y \in \mathbb{Z} \]

\[ (29) \]

Figure 15. Two Level Discrete Wavelet Decomposition of Secret Image

Where \( f(\omega_x, \omega_y) \) is 8 * 8 image block, \( K(\omega_x, \omega_y) \) is quantization step which, for JPEG quality is given by:

\[ K(\omega_x, \omega_y) = \begin{cases} \max \left( \frac{200 - 2Q}{100}, \frac{Q T(\omega_x, \omega_y)}{2} + \frac{1}{2} \right), & 50 \leq Q \leq 100 \\ \frac{50}{Q}, & 0 \leq Q \leq 50 \end{cases} \]

\[ (30) \]

Where \( Q T(\omega_x, \omega_y) \) is quantization Table. Some researchers, such as Li and Wang (2007), have modified QT and hidden information in the middle frequency coefficient. According to results obtained by Chang, et al., (2002); Fard, et al., (2006); Hashad, et al., (2005), change in frequency domain will be less visible in cover image than spatial domain. On the other hand, spatial-domain provides large capacity for hiding information in comparison to frequency domain (Kong, et al., 2007). Among different frequency domain algorithms (DCT, DWT, DFT, RDWT), DWT is more common. Many wavelet-based watermarking algorithms are introduced as yet. Different filter banks and various decomposition levels can be used in DWT. As an example, some researchers have exploited Harr function (Peng and Liu, 2008; Vahedi, et al., 2007a,b) and others have used Daubechies-2 (Vahedi, et al., 2007b). Yet, some other researchers have used different levels of decomposition. For example, Zh. Peng, et al., exploit 1-level decomposition (Peng and Liu, 2008) and (Kundur and Hatzinakos, 2004; Agreste, et al., 2007) have used three level, and (Reddy and Chatterji, 2005; Vahedi, et al., 2007a) have applied four-level decompositions. E.Vahedi, et al., (2007c) showed that an appropriate selection of wavelet filter banks, decomposition level and embedding sub-space had an important effect on the transparency and robustness of the final watermarked image. Yang, et al., (2011) used frequency domain only as compression method. In their algorithm, first, each secret image is divided into four sub bands (LL, LH, HL, HH) using Haar discreet wavelet transform (Figure 15) (DWT) and LL sub-band is used for sharing among participants. LL contains maximum energy. After performing DWT, the LL sub-band is shared in natural images by Shamir’s polynomial algorithm Lin and Tsai (2004). Figure 16 illustrates the example of C. Huei, et al.
The disadvantage of this method is that two location maps are needed to maintain the reversibility: one for the shuffle pixels generated by pseudo-random generator (PRNG) and another for recording position of pixels with grayscale value 251-255. Other drawbacks of this scheme are that stego-images are noise-like. So, it attracts attacker attention.

![Image of Lena secret image and discrete Haar LL subband]

**Figure 16. An Example using Lena to Present the (2;4) Threshold Condition Lin and Tsai (2004)**

Peng and Liu (2008) shared color secret image in frequency domain using DWT. Their scheme involves two phases: watermark embedding and watermark extraction phase. In the first phase, color image is converted from RGB into YCbCr. Then, special sampling is created from YCbCr color space. Next, discrete wavelet transform is applied for sampling plane to extract features. Afterwards, according to the feature and watermark a principal image is generated. Finally, discrete wavelet transform is used to embed this principal share image in the color host image to produce the watermarked image. In extracting phase, all watermarked images are used to retrieve principal share image. Then, an expanded watermark is reconstructed using the features of the manipulated image and the retrieved principal share image. At last, the watermark is recovered and can be used to analyze the copyright. Secret sharing in frequency domain can be useful in copyright protection and watermarking. As an example, in copyright protection, instead of embedding watermark in host image, it is embedded in two shares (ownership and master). Ownership share is created from secret image and master share is created from host image Wu and Sun (2013a). The master share construction can be divided into two categories: master share construction in spatial domain and master share construction in frequency domain. For instance, in frequency domain, Hsieh and Huang S. Hsieh (2009) exploited a method based on DWT and VC technique. First, host image is divided into 8x8 non-overlapping blocks. Then two scales DWT is applied to these blocks and LL sub band of each block is obtained. Then, mean value of each block is calculated. So, according to a predefined value between the mean value and LL coefficient, the corresponding block in master share is generated. An example of this scheme is presented in Figure 17.

Recently, Wu and Sun (2013b) proposed a method for secret sharing based on cellular automata and discrete wavelet transform. Their scheme could authenticate manipulated image and repair the manipulated area by hidden information. Their
scheme will be described in three phases: setup phase, sharing and embedding phase and authentication phase. In the first phase, some parameters such as: bit depth=8, automata radii (r = 3),k = 3 and n = 3 are fixed. Afterward, for constructing linear memory cellular automata, a number (rule number) is selected randomly from $[0;2^{2r+1} - 2]$. Then, local transition function is constructed by:

$$ a_i^{(r+1)} = f_{x_1}(N_i^{(r)}) + f_{x_{r+1}}(N_i^{(r-1)}) + a_i^{(r-2)} \mod 2 $$

(31)

Where $0 \leq i \leq 7$.

Figure 17. Copyright Protection by Secret Sharing, (a) Host Image, (b) Binary Watermark, (c) Master Share, (d) Ownership Share, (e) Revealed Watermark

Figure 18. Diagram of Creating Partner Image

In the sharing and embedding phase, 1-scale DWT is applied to the secret image and coarse sub-bond is selected (LL$_1$). LL$_1$ is scrambled and after bit extraction, it is rearranged to create partner image. Both partner image and secret image are separated to non-overlapping blocks. Those blocks are the initial configuration of 3-order linear memory cellular automata. By evolving Equation 31, share bits are generated. Then, authentication bit and share bits are embedded in 2 LSB with 2 4 blocks of cover image to form Stego-image. See Figure 18.

In revealing and repairing phase, when all partner is present, the secret image can be reconstructed. The share bits are retrieved from stego-image. These shares are initial configuration in reverse LMCA. By evolving inverse LMCA, the secret image and partner image can be reconstructed. Then, reconstructed pixels in partner image is
scrambled and rearranged to form approximation image. Finally, inverse DWT is applied to approximation image to form the copy of secret image. If each pixels of original secret image is manipulated or damaged, then this pixel is replaced by corresponding pixels in copied secret image. Drawback of this scheme is that, cover image is four times bigger than secret image. So, in some places which have network constraint it might be a problem. Eslami, et al., scheme (Eslami, et al., 2010) also has this drawback. In comparisons to Chang, et al., scheme (Chang, et al., 2008), Eslami, et al., scheme and Chang, et al., scheme (Chang, et al., 2011), X Wu and W Sun's has higher stego image quality. Also, probability of fake block detection of this scheme is higher. In Table 3, Chang, et al., Eslami, et al., and X Wu and W Sun's schemes are compared based on main feature. This table showed that X Wu and W Sun's schemes are the same as Eslami, et al., scheme in number of authentication bits, Fake block, detection probability, number of LSB embedding, block size and computational cost of decryption. But, Eslami, et al., could not authenticate and remedy fake stego-image. So, Xu and W Sun's method outperforms previous algorithms in the time of writing this article.

Table 3. Comparison between Some Secret Sharing Schemes in Authentication and Remedy Ability

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of authentication bits</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Fake block detection probability</td>
<td>$\frac{1}{8}$</td>
<td>$\frac{2}{8}$</td>
<td>$\frac{1}{8}$</td>
<td>$\frac{2}{8}$</td>
</tr>
<tr>
<td>Number of LSB Embedding</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Block size</td>
<td>$2 \times 2$</td>
<td>$2 \times 2$</td>
<td>$2 \times 2$</td>
<td>$2 \times 2$</td>
</tr>
<tr>
<td>Computational cost of decryption</td>
<td>$O(n \log^2 n)$</td>
<td>$O(n)$</td>
<td>$O(n \log^2 n)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Remedy ability</td>
<td>NO</td>
<td>NO</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

According to X Wu and W Sun's scheme, a copy of image (partner image) is created by approximation image (LL sub-bond of DWT). Both partner image and shared image are embedded in cover image. When stego-images are damaged, the damaged area can be recovered by taking inverse DWT of approximation image and replacing it with faked original secret image. So, this scheme is robust against attackers. X Wu and W Sun showed that if the area spanning 60% of stego-image is damaged, this method will be able to recover the secret image. Figure 19 demonstrated an example of recovering fake stego-image in X Wu and W Sun's scheme. This figure showed that manipulated blocks of stego-image can be detected completely. Also, PSNR of repaired secret image is obviously good.
5. Analysis and Recommendation

As previously mentioned, PSNR can be used as a performance measure for image distortion. Usually, PSNR values of less than 30dB indicate low quality, i.e., distortion caused by embedding can be visible. Also, stego-image with PSNR of 40dB and above is known to have high quality. Figure 20 shows different PSNR values for different algorithms. The selected images are Bridge, Elaine, Goldhill, Lake, Peppers and Earth. Image size is 512*512. Average PSNR of these images showed that the quality of X Wu and W Sun's algorithm is better than other algorithms. The LSB embedding strategy is an efficient approach toward hiding high capacity data in cover image. But, robustness of the conceal data is scarified. Hiding data in transform domain (e.g. embedding in discrete cosines transform or discrete wavelet transform) is more robust against noise addition and other attacks. But, more issues such as capacity of embedding and visual quality must be further studied. Table 4 compares mentioned algorithms based on following factor:

- Average PSNR range.
- Authentication and remedy ability.
- Algorithm which is used for creating share values.
Embedding method.
Year of proposed.

Table 4. Compare Different Algorithms of Secret Image Sharing

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Average PSNR range</th>
<th>Remedy ability</th>
<th>Based on</th>
<th>Embedding method</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gohari (1979)</td>
<td>-</td>
<td>No</td>
<td>Lagrange polynomial interpolation</td>
<td>LSB spatial domain</td>
<td>1979</td>
</tr>
<tr>
<td>Lin and Tsai (2004)</td>
<td>34 – 30</td>
<td>Yes, only authentication</td>
<td>Lagrange interpolation, threshold scheme</td>
<td>spatial domain</td>
<td>2004</td>
</tr>
<tr>
<td>Yang et al. (2007)</td>
<td>29 – 41</td>
<td>Yes, prevent dishonest participants from obscuring</td>
<td>Lagrange interpolation based on Galois Field</td>
<td>spatial domain</td>
<td>2007</td>
</tr>
<tr>
<td>Chang et al. (2008)</td>
<td>38 – 41</td>
<td>4-bit authentication with no remedy</td>
<td>Polynomial interpolation, employ all k secrets as coefficients in ( \frac{1}{n} ) - degree polynomial</td>
<td>spatial domain</td>
<td>2008</td>
</tr>
<tr>
<td>Elam et al. (2010)</td>
<td>47</td>
<td>Yes, Authentication only</td>
<td>Memory cellular automata</td>
<td>Spatial domain</td>
<td>2010</td>
</tr>
<tr>
<td>Wu et al. (2011)</td>
<td>43</td>
<td>Yes, Authentication only</td>
<td>Lagrange interpolation with optimal pixel adjustment</td>
<td>Spatial domain</td>
<td>2011</td>
</tr>
<tr>
<td>Elam and Ahmed (2011)</td>
<td>48</td>
<td>Yes, authentication only</td>
<td>Lagrange polynomial interpolation, dynamic embedding</td>
<td>spatial domain</td>
<td>2011</td>
</tr>
<tr>
<td>Al-Qershi and Khow (2011)</td>
<td>-</td>
<td>No</td>
<td>Difference expansion</td>
<td>Frequency domain</td>
<td>2011</td>
</tr>
<tr>
<td>Wu and Sun (2012)</td>
<td>37 – 45</td>
<td>Yes, Authentication and remedy</td>
<td>Two dimensional memory cellular automata</td>
<td>Spatial domain</td>
<td>2012</td>
</tr>
<tr>
<td>Chang et al. (2013)</td>
<td>45 – 46</td>
<td>Yes, Authentication and remedy</td>
<td>Lagrange polynomial interpolation</td>
<td>Dynamic domain</td>
<td>2011</td>
</tr>
<tr>
<td>Wu and Sun (2012)</td>
<td>47</td>
<td>Yes, Authentication and remedy, noise sprinkling resistance</td>
<td>Memory cellular automata</td>
<td>Dynamic domain</td>
<td>2012</td>
</tr>
</tbody>
</table>

6. Conclusions and Summary

This paper presented a review on the recent secret image sharing algorithms. Secret image sharing in this paper was divided into two categories: spatial domain and transform domain. Reported experimental result showed that embedding in spatial domain had more capacity in comparison to transform domain. But, embedding in transform domain was more secure and resistant to stegoanalysis attacks. Because of embedding capacity, most of secret image sharing algorithms are based on spatial domain. Transform domain is useful in watermark embedding, because, watermark image size is very smaller than cover image size. There are several ways to reducing the bits required for embedding in cover image. For example, jpeg compression can be used to reduce size of secret image. Also, dynamic embedding can increase embedding capacity. In short, there must be a trade-off between embedding capacity and robustness.

This paper also explained the differences between cellular automata and La-grange polynomial. The cellular automata had less time complexity. Change in the structure of cellular automata is easier than Lagrange polynomial. Also, 1D and 2D cellular automata were introduced in this paper. Authentication bits can be vary. The higher the number of bits, the higher the authentication ability and security would be. In contrast, image quality can be low and distortion can be high.
References


1852-1863.


