Secret Image Sharing based on Wavelet Transform

Ashwaq T. Hashim a, Dr. Loay E. George b, *

a Computer Science Department, Science College, Babylon University, Iraq
b Computer Science Department, Science College, Baghdad University, Iraq

Abstract

Visual cryptography is a secure technique enable distributing sensitive visual data to participant through public communication channels. In visual cryptography a secret image is cryptographically encrypted into some shares (transparencies) such that the generated secure shares do not reveal any information without they are not combined in prescribed way.

The main challenges facing secure image sharing tasks are the increase of sharing volume and sharing-control flexibility. In this paper a new scheme based on wavelet transform is developed to perform packet secret image sharing in the case when the secret image is a color image with enough sharing control features and as minimum as possible shares sizes. The main signal decomposition attribute of the transform is utilized to divide the secret image into uncorrelated shares, and each share is, separately, encrypted.

Keywords: Visual cryptography; Secret image sharing; Wavelet transform; Image compression

1. Introduction

Due to the rapid progress in Internet and digital imaging technology, the use of digital images is increasing rapidly. The issue of privacy protection of images became an necessary demand especially in the image distribution applications. Visual cryptography and Secret Image Sharing is one of the novel tasks to protect image privacy against unauthorized data access. Nowadays it is considered as a key solution for secure dissemination of sensitive information.

Numerous approaches have been developed for image sharing. Firstly Shamir [1] and Blakley [2] first proposed a concept of secret sharing called the \((r, m)\) threshold scheme. In their scheme, a secret is shared by \(m\) shadows and any \(r\) shadows, where \(r \leq m\) can be used to reveal the secret while with less than \(r\) shadows the information about the secret cannot be obtained. Thien and Lin [3] in 2002, extended a secret image sharing method based on Shamir’s \((r, m)\) threshold scheme. Their method permutes a secret image first to de-correlate pixels and then incorporates the \((r, m)\) threshold scheme to process the image pixel wise or pattern wise in the spatial domain sequentially. Each generated shadow is \(1/r\) the size of the original image.

Instead of using permutation to de-correlate pixels prior to applying the \((r, m)\) threshold scheme as in [3], the proposed system first was applied Biorthogonal wavelet transform and then process transform coefficients in a preprocessing stage to de-correlate pixels (coefficients), get excellent energy compaction maps and increase security. Each generated shadow by proposed system is proximately \(1/3r\) of the size of the original color image.

Due to the great research effort spent in reducing the shadow size each shadow size further has been reduced, Lin and Tsai [4] in 2003, transformed a secret image into the frequency domain by applying the discrete cosine transform (DCT). Then all the DCT coefficients except the first 10 lower frequency ones are discarded. And the values of the 2nd through the 10th coefficients are disarranged in such a way that they cannot be recovered without the first coefficient and that the inverse DCT of them cannot reveal the details of the original image. Finally, the first coefficient is encoded into a number of shares for a group of secret sharing participants and the remaining nine manipulated coefficients are allowed to be accessible to the public. In this secret image scheme, this reduces the size of the original secret image with degrading the image quality. If higher image quality is desired, it may keep more than 10
coefficient. The security of the retrieval of the original secret image by brute force attack depend only on first coefficient $C_1$ which has been encrypted with the Shamir secret sharing scheme into $n$ shares.

In 2006, Wang and Su [5], proposed a method of secret image sharing by applying the image difference and the algorithm of Huffman coding in the sharing process. In the method proposed in this study, the difference image of the secret image is encoded using Huffman coding scheme, and the arithmetic calculations of the sharing functions are evaluated in a power-of-two Galois Field GF(2). In this scheme the generated shadow image is about 40% smaller than that of the method in [3]. Chin and Ching [6] in 2006, present a study on an image sharing method applying the integer multiwavelet transform and Shamir’s $(r, m)$ threshold scheme that provide highly compact shadows for real-time progressive transmission. In 2007, Chin and Ching [7] present a new image sharing method, based on the reversible integer-to-integer (TI) wavelet transform, that provides highly compact shadows for real-time progressive transmission. This method, working in the wavelet domain, processes the transform coefficients in each subband, divides each of the resulting combination coefficients into $m$ shadows, and allows recovery of the complete secret image by using any $r$ or more shadows ($r \leq m$). This method without coding has larger shadow images than those of method that has been coded prior to inputting to the sharing phase. This method was also encoded either with Huffman coding or with arithmetic coding before the data input to the sharing phase.

In 2008, Chin-Chen Chang et al [8], combine Chang and Wu’s gradual search algorithm for a single bitmap BTC (GSBT) to compress a secret color image and then Shamir’s $(r, m)$ threshold concept to propose a novel secret color image sharing scheme that generates smaller shadows. In this scheme each generated shadow is proximately $1/2r$ of the size of the original color image with PSNR 29 dB while each generated shadow by proposed system is proximately $1/3r$ of the size of the original color image with PSNR 34 dB. Liu et al [9] in 2008, introduced $(r, m)$-threshold scheme for image sharing; it is based on transform domain coefficients operations. This scheme is effective and perfectly secure due to the DFRNT exploited. However, in this scheme all of the shadows are of the same size as that of the secret image and thus much storage space is required and more transmission time is spent.

In 2009, Zhenfei Zhao et al [10] proposed a method that incorporates secret image sharing with progressive transmission based on the integer discrete cosine transform (IntDCT). In this progressive shares transmission system, a coarse version of the secret image can be encrypted with the first part of the received shares and this coarse image can be refined in the successive stages. If the decrypted image quality is good enough, shares transmission can be interrupted. Zhao and et al [11] proposed, in 2010, an improved image secret scheme based on the discrete fractional random transform, in this $(r, m)$ threshold prototype, the shadow size is reduced to $1/r$ of the secret image.

Yang, et al [12] proposed, in 2011, a fast secret image sharing based on Haar wavelet transform and Shamir’s method. They employ discrete Haar wavelet transform to reduce the secret image to its quarter size firstly (i.e., 1-level LL subband). Then, the modified Shamir’s algorithm is applied only on this LL subband to generate the shadow images. Chin and Ching [13] in 2011, present a new secret image sharing method based on the multiwavelet transform and the SPIHT algorithm. The method extends [6,7] to a more general framework and improves their performances. It utilizes a data structure rearrangement process for multiwavelet transform coefficients in subbands to adapt their structure efficiently for the SPIHT encoding. Hemrajani and et al [14], in 2012, have proposed a variable length Symmetric Key based Visual Cryptographic Scheme for color images; a secret key is used to encrypt the image and the division of encrypted image is done using Random Number, without the secret key the original image can't be decrypted. Here the secret key ensures the security of the scheme and visual cryptography is used to break the image into number of shares.

In comparison to these existing methods, all previous works utilized Shamir’s $(r, m)$ threshold concept to generates shadows after preprocessing the original secret image by various techniques. In proposed system, a linear system has been used to generate $m$ shares and the security of the retrieval of the original secret image depend on the $r$ secret coefficients associates with each shares and $r$ secret value from transformed, compressed and ciphered data of original secret image. The proposed method, working in the wavelet domain, has the advantage of having small shadow images and secure transmission at the same time.

2. Proposed Secret Image Sharing System

The color image occupy more space and consume more bandwidth compared to grayscale and binary images since reducing the image size of color images is very important for efficient transmission and storage of images.

During the design phase of the proposed system the main concert was focused on:

- Mainly through better energy compaction property of wavelet biorthogonal tap 9-7 transform, wavelet based coding has been provided great improving in image quality at high compression ratios mainly [15].
- All subbands of the wavelet transform (i.e., LL, LH, HL, HH) for all color components (like YUV) are used to generate shares because the encrypted images might be a document where edges is the most important part of the signal; in such case the subbands LH, HL, and HH will be energetic and can't be neglected.
- The adaptive Shift-encoder is exploited as entropy encoder in order to make the representing values of the resulted codeword with minimum bits. Shift
encoding scheme is exploited because of needing a little amount of overhead information.

- Bitwise diffusers are used to prune the existing bits significance in wavelet coefficients, this step handles the effect of transform localization attribute.
- Randomize the secret information lay into each share using a secret key.

As shown in figure (1), the system layout consists of following steps:

**Step1:** Transform the color components (RGB) of the input color image into color space components with less correlated (like YUV) to reduce the spectral redundancy.

**Step2:** Down sample the chrominance components (U, V) by 2.

**Step3:** Pass the Y component and the down sampled (V’ and U’) de-correlated color components to the proposed Visual Cryptography system (VC).

**Step4:** The resulted shares from VC is encrypted, using the proposed encryption system; this step will the security immunity of the resulted shares.

**Fig.1. Proposed system**

**2.1 Proposed Visual Cryptography (VC)**

This part was carefully designed to ensure the generated shares as secure as possible with minimum size. As shown in figure (2), the proposed Visual Cryptography (VC) consists of the following steps:

**Step1:** Perform tap 9-7 wavelet transform on Y and the down sampled V’, U’ chromatic bands. The transform decomposes each color band data to the known wavelet subbands (LL, LH, HL, HH).

**Step2:** A uniform scalar quantization scheme is applied upon the transformed data to quantize the approximation (LL) subband (i.e., rounding only) and the detail subbands (uniform quantization); where each component (LH, HL and HH), is quantized using different quantization step (q1, q2, q3, and q4 respectively). The quantization steps values have been choices due to the subjective significance of each wavelet band.

**Step3:** After the quantization step the quantized subband has been divided into 8x8 blocks; then a quadtree-based search is performed to check the emptiness of the tested blocks, the search is repeated on its four daughter quadrants in a hierarchical manner until reaching quadrants have size (2x2). For 2x2 non empty blocks their contents and the quadtree coding sequence are saved in buffers.

**Step4:** The quantized approximation subbands (LL) for Y, U’, and V’ are modulated by determining the differences adjacent coefficients:

\[ S_i = S_{i-1} - S_{i-2} \quad \text{for } i=1,\ldots,n-1 \]  

where n is the length of subband.

**Step5:** Entropy encoding (i.e., using shift encoder) is applied to encode the modulated (LL) and also the contents of buffer which contain the nonempty (2x2) blocks. The following steps are applied to get high compression gain

**Step5-1:** Positive mapping: by this stage the coding complexity that result from the existence of positive/negative values can be overcome. These quantized coefficients values are converted to be always positive, by performing the following equation:

\[ X_i = \begin{cases} 2X_i & \text{if } X_i \geq 0 \\ -2X_i - 1 & \text{if } X_i < 0 \end{cases} \]  

where \( X_i \) is the \( i \)th element. by the above equation all the positive values have became even while the negative values are converted to become odd.

**Step5-2:** The adaptation of the shift-encoder as entropy encoder because of:

a. ease of implementation in both encoding and decoding.

b. the size of its corresponding overhead information is small (i.e., only two numbers; which are the length of short and long codewords in terms of bits).

To determine the optimal values of the short (\( n_s \)) and long (\( n_l \)) codewords the following steps were applied:

a. Let \( n_l = \left\lfloor \log_2(L) \right\rfloor \) where \( \left\lfloor x \right\rfloor \) means the smallest integer value higher than x, \( L \) is the highest value found in the stream of collected coefficients (after making map-to-position task).

b. Find the \( n_s \) value that leads to lowest possible values for \( n_r \), where

\[ n_r = n_l \sum_{i=0}^{b} \text{His}(i) + n_s \sum_{i=n}^{b} \text{His}(i) \]  

Save the output codewords in the binary compression stream of bits.

**Step6:** The compression stream is randomly diffused to prune the bits significance of the wavelet coefficients. Different diffuser have been applied to diffuse (1) LL compressed subbands, (2) the compressed output of
quadtrees of the nonempty blocks of detail subbands and (3) to the sequence of the empty and nonempty blocks of the quadtree.

2.1.1 Design of Proposed Diffusers
The basic ingredients of modern fast software encryption schemes are the primitive bitwise computer instructions like ROTATE, ADD, XOR etc. Different subsets of such operations will yield an interesting variety of different permutation groups, (e.g. symmetric groups). For example a simple pair of ROTATE and an ADDITION module is powerful enough to generate every possible hand, any possible combination of ROTATE and XOR operations can only produce a subset of at most $n \times 2^n$ functions within the symmetric group of order n! [16].

I. Diffuser
This reversible diffuser is applied to the modulated LL subbands coefficients. It is required to provide the necessary diffusion, It uses a mixture of operations consists of different algebraic group: XOR, addition, and multiplication. Diffuser1 implies set of operation on sets consists of four 32-bit words, it uses the following six basic operations.

\begin{align*}
+ & : \text{for integer addition modulo } 2^w \\
- & : \text{for integer subtraction modulo } 2^w \\
\oplus & : \text{bitwise exclusive-or-of } w\text{-bit} \\
\times & : \text{integer multiplication modulo } 2^w \\
<<< & : \text{rotate the } w\text{-bit word a to the left} \\
>>> & : \text{rotate the } w\text{-bit word to the right}
\end{align*}

For each input block $X$ of length 16 bytes the following operations are performed:

**Step1:** Divide $X$ into four 32-bits quadratic $A$, $B$, $C$, and $D$.

**Step2:** Then perform the following operations:

\begin{align*}
x1 &= (A \times (2A + 1)) << 5 \\
y1 &= (C \times (2C + 1)) << 5
\end{align*}
The output from shift encoder represents the compressed secret information; it will be distributed randomly into n shares. By using a pseudo random generator dependent on a secret key and a timestamp, the q segments of secret information will be permuted randomly.

Figure (3) presents the steps taken to apply shuffling algorithm. This process consists of the following steps:

**Step1:** Initialize b as a sequence of length n; such as \( b[i] = i \mod q, \text{ for } i = 1 \ldots n \)

**Step2:** Let \( j = R1 \)

**Step3:** for \( i = n-1 \ldots 1 \)

\[ j = (R2 \times j + R3) \mod i \]

swap \( b(i), b(j) \)

where \( R_i, R_2 \), and \( R_3 \) large prime numbers they were generated using the proposed key generation function shown in figure (4).

The inputs to \( R_i \) function are: (i) a secret key of length 48 bytes \( (R_1, R_2, R_3) \), (ii) date and time; while the outputs are three large prime numbers \( R_i, R_2 \), and \( R_3 \) each of which is 16 bytes. The following steps show the generation steps of \( R_i, R_2 \), and \( R_3 \):

\[ R1 = x4 \oplus x5 \oplus x10 \oplus x15, \quad R2 = x3 \oplus x6 \oplus x9 \oplus x16, \quad R3 = x1 \oplus x2 \oplus x8 \oplus x13 \]

\[ R1 = y1 \oplus y2 \oplus y3 \oplus y4 \oplus y5 \oplus y6 \oplus y7 \oplus y8 \oplus y9 \oplus y10 \]

\[ R2 = y11 \oplus y12 \oplus y13 \oplus y14 \]

\[ R3 = y15 \oplus y16 \]

**2.1.2 The Proposed Random Generation Function**

Let Date \((Y, M, D)\)

\[ m_1 = Y; \quad m_2 = M; \quad m_3 = D; \quad m_4 = Y \oplus M \oplus D \]
Step2: Let Time (H, Min, S)
\[ n_1=H; n_2=\text{Min}; n_3=S; n_4=H \wedge \text{Min} \wedge S \]

Step3: The secret key K will be permuted depending on Date and Time in F function it has 128-bit input A and 128-bit output D. It Adopts "byte transposition" and the subkey (K_i, K_j) to control data rotations.

- Let \( K_i = (m_1, m_2, m_3, m_4) \), and \( K_j = (n_1, n_2, n_3, n_4) \)

the function \( D = F(A, K_i | K_j) \) is consists of the following steps:

  Right rotation: \( b_j = a_j \gg m_j \) for \( j=1,...,4 \).
  Byte transposition: \( c_j = b_j \) for \( j=1,...,4 \).
  Left rotation: \( d_j = c_j << n_j \) for \( j=1,...,4 \).

Figure (5) illustrates the Permutation function \( F_j \)

2.1. The Proposed Sharing Scheme

Most of proposed methods published in the literature imply the idea of dividing the secret image data into a set of non-overlapping segments; each segment has \( q \) bytes, and usually its content is represented using \( q-1 \) degree polynomial [2, 10, 12]. In this paper, the proposed sharing system is based on using a set of linear equations; such that each \( i^{th} \) share has a secret set of \( a_{ij} \) (where, \( i \in [1, n] \) and \( j \in [1, q] \), \( n \) is the total number of shares and \( q \) is the minimum number of required shared to retrieve the image data). In other words, for a block of image data \( \{V_j\}_{j=1}^q \), the \( i^{th} \) share is computed using the following linear equation:

\[
Sh_i = \sum_{j=1}^q a_{ij} V_j
\]

Where, \( Sh_i \) is the \( i^{th} \) generated share for the block \( V() \), \( a_{ij} \) is the \( j^{th} \) coefficient belong to the linear equation representing the \( i^{th} \) share. So in case of collecting any \( q \) shares from the total \( n \) shares (i.e., \( \{Sh_i | k=1...q\} \)), then the inverse matrix of \( A=\{a_{ij} \} \ i,j=1...q \) could be used to retrieve the exact values of \( V() \), that is:

\[
V = A^{-1} Sh
\]

The proposed method can be divided into the sharing phase and the reveal phase. Details about the two phases of our proposed algorithm are described below.

Beside to utilization of linear equation the Modula algebra is used to overcome the size increase of the overall shares size. So, instead of equation (1) the adopted share generation equation is:

\[
S_i = \sum_{j=1}^q (a_{ij} V_j) \mod 256
\]

Where, \( i=1...n \) and \( j=1...q \). According to equation (3), the range of shares values \( S_i \) is \([0..255]\). The above equation could be rewritten as in the following:

\[
\sum_{j=1}^q a_{ij} V_j = S_i + 256p_i
\]

Where, \( p_i \) is an integer number its value will not registered as a part of share values, and during the retrieval stage their values will be compensated according to certain integer division based rules.

The reveal phase is the inverse coding process of the sharing phase. In this phase any \( q \) different shares, taken from the total \( n \) shares, are collected for decryption. These \( q \) shares are used to construct \( q \) simultaneous linear equations set (i.e. one for each share), and thereby the secret bytes \( \{V_j\}_{j=1}^q \) can be obtained by solving these linear equations set. If less than \( q \) of simultaneous linear equations are collected, the linear equations cannot be solved to retrieve the secret bytes \( \{V_i\} \).

Since the proposed system uses Modula algebra with base 256 to reduce the range of the generated shares and keep them within the range \([0, 255]\), so the algebra needed to recover of secret bytes \( \{V()\} \) should take into consideration the imposed range restriction of the shares.

The following steps have been adopted for the recovery process:

Step1: Take the shares whose corresponding indexes are \( \{n_j, n_{j}, ..., n_j\} \); such that only one secret byte value is taken from any chosen share (i.e., \( \{S_{n_j} | m=1,2,...q\} \)).

Step2: Construct the coefficients matrix, \( A'() \), of the corresponding linear equations, that is

\[
a'_{mk} = a_{n_{mk}}
\]

Where, \( a'_{mk} \in A' \), \( a_{n_{mk}} \in A \), \( m=1,2,...,q \) and \( k=1,2,...,q \).

Step3: Determine the determinant value of \( A \) (i.e., \( D=\text{det}(A) \)), and the corresponding complementary matrix \( C \); such that for all values of \( j \) (i.e., \( j \in [1, q] \)) the following condition is satisfied:

\[
\frac{\sum_{j=1}^q a_{ij} C_j}{\sum_{j=1}^q a_{ij} C_j} = D
\]
Here, the matrix element \( C_{ij} \) is equal to the determinant of the reduced matrix \( C \) whose \( i^{th} \) row and \( j^{th} \) column are removed multiplied by the factor \((-1)^{i+j}\).

**Step4:** The values of the retrieved secret bytes \( \{V'_j\} \) could be determined using:

\[
V'_j = \frac{L}{D} \left( \sum_{i=1}^{n} C_{ij}S_{m} \right) + w_j \quad (11)
\]

Where, \( w_j \) is an integer number its value is multiples of 256, such that:

\[
w_j = 256 \sum_{i=1}^{n} C_{ij}p_i \quad (12)
\]

According to the above equations for shares generation \( \{S_i/\} \) and the secret bytes retrieval \( \{V/F\} \), the following two remarks are taken into consideration:

(a) The value of \( D \) should always kept non zero to ensure the applicability of equation (11); so, the generation process of all sharing matrix coefficients (i.e., \( a_{ij}\in A \; i=1...n, j=1...q \) ) should take into consideration that any combination \( (q\times q) \) of \( a \) 's coefficients should not lead to zero.

(b) Since the values of \( p \)'s (see equation 8), will not registered as part of the share data, so the values of \( w \)'s could not be determined directly from equation 12. To handle this problem the exhaustive test for the all possible values of \( w_j (j\in[1,q]) \), as multiples of 256, are tried. Here, the correct set of \( w_j \) values is selected when all values of the retrieved \( \{V'_j\} \) are integer. In other words; the values of all numerators of equation (11) that are multiples of the denominator (i.e., \( D \) ) value are considered during the test.

### 3. Experimental Results and Discussion

Various experiments have been carried out to assess the performance of the proposed algorithm. Four color scale images, "Lena", "Baboon", "Peppers" and "Lichtenstein" of size \( 512\times512 \) were used as secret images. These test images are shown in Figure(6).

Table (1) lists the number of bytes spend to encode each image for various states of quantization step, the quantization step \( q_1 \) is applied on the LL subband component , \( q_2 \) is the quantization step applied on LH,HL bands; while for the HH band the quantization step is set to \((1.4* q_2)\).

The peak signal to noise ratio (PSNR), defined in equation (7), was adapted to evaluate the visual fidelity of the constructed image in comparison with the original image [17] as shown in table (2):

\[
PSNR = 10 \log_{10} \frac{255^2}{\text{MSE}} (\text{dB})
\]

**Table 1:** The number of bytes spend to encode each image

<table>
<thead>
<tr>
<th>Quantization Step</th>
<th>Number of Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_1 )</td>
<td>Lena</td>
</tr>
<tr>
<td>2</td>
<td>284591</td>
</tr>
<tr>
<td>10</td>
<td>1346725</td>
</tr>
<tr>
<td>25</td>
<td>333619</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>281601</td>
</tr>
<tr>
<td>25</td>
<td>276134</td>
</tr>
<tr>
<td>20</td>
<td>273591</td>
</tr>
<tr>
<td>25</td>
<td>276581</td>
</tr>
</tbody>
</table>

where MSE is the mean squared error between the original image and the retrieved image, it is given by [17]:

\[
\text{MSE} = \frac{1}{mn}\sum_{x=1}^{m}\sum_{y=1}^{n}(X(x,y) - X'(x,y))^2 \quad (14)
\]

Figure 7 shows an example of the proposed system output, the sharing scheme was set \( (n=4, q=2) \). The original size of Lena is 786,432 bytes and after compression the compressed size become 284591 bytes (with \( q_1=2 \) and \( q_2=5 \)), and the size of each share become 142,296 byte.

Table 3 shows the time comparison that required to reconstruct the secret image "Lena" (with \( q_1=2 \) and \( q_2=5 \)), from a selected pair of shares using different range of secret coefficients, \( a \) , used by each shares (i.e., share1 used the coefficient values \( a_{11}=131, a_{12}=137 \) and share2 used \( a_{11}=139, a_{22}=134 \) as their own coefficients receptively, when share1 and share2 are selected to reconstruct secret image, the required average time to reconstruct the original image from the two shares is 13.49 second.
Table 2 - A comparison between the MSE and PSNR for different cases of quantization steps

<table>
<thead>
<tr>
<th>Quantization Step</th>
<th>Lena</th>
<th>Baboon</th>
<th>Peppers</th>
<th>Liechtenstein</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td>q2</td>
<td>PSNR</td>
<td>MSE</td>
<td>PSNR</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>34.498</td>
<td>23.077</td>
<td>35.981</td>
<td>16.404</td>
</tr>
<tr>
<td>15</td>
<td>34.375</td>
<td>23.741</td>
<td>35.738</td>
<td>17.345</td>
</tr>
<tr>
<td>20</td>
<td>34.327</td>
<td>24.006</td>
<td>35.573</td>
<td>18.019</td>
</tr>
<tr>
<td>25</td>
<td>34.314</td>
<td>24.080</td>
<td>35.499</td>
<td>18.329</td>
</tr>
<tr>
<td>30</td>
<td>34.314</td>
<td>24.080</td>
<td>35.480</td>
<td>18.402</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>34.458</td>
<td>23.295</td>
<td>35.894</td>
<td>16.736</td>
</tr>
<tr>
<td>10</td>
<td>34.169</td>
<td>24.897</td>
<td>35.531</td>
<td>18.195</td>
</tr>
<tr>
<td>15</td>
<td>34.055</td>
<td>25.559</td>
<td>35.306</td>
<td>19.160</td>
</tr>
<tr>
<td>20</td>
<td>34.011</td>
<td>25.820</td>
<td>35.157</td>
<td>19.832</td>
</tr>
<tr>
<td>25</td>
<td>33.998</td>
<td>25.893</td>
<td>35.089</td>
<td>20.141</td>
</tr>
<tr>
<td>30</td>
<td>33.998</td>
<td>25.893</td>
<td>35.074</td>
<td>20.214</td>
</tr>
</tbody>
</table>

Table 3 - Time comparison that required to reconstruct secret image when different ranges of a( ) coefficient values of is adapted values

<table>
<thead>
<tr>
<th>No of Shares</th>
<th>Range of coefficients</th>
<th>Time in Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10-39</td>
<td>0.18-0.90</td>
</tr>
<tr>
<td></td>
<td>40-69</td>
<td>1.48-3.12</td>
</tr>
<tr>
<td></td>
<td>70-99</td>
<td>4.07-6.79</td>
</tr>
<tr>
<td></td>
<td>100-128</td>
<td>8.08-11.51</td>
</tr>
<tr>
<td></td>
<td>130-159</td>
<td>13.49-17.49</td>
</tr>
<tr>
<td></td>
<td>160-189</td>
<td>20.29-25.83</td>
</tr>
<tr>
<td></td>
<td>190-219</td>
<td>28.86-34.99</td>
</tr>
<tr>
<td></td>
<td>220-255</td>
<td>38.22-49.10</td>
</tr>
<tr>
<td>3</td>
<td>10-39</td>
<td>17.34-105.41</td>
</tr>
<tr>
<td></td>
<td>40-69</td>
<td>192.98-785.79</td>
</tr>
<tr>
<td></td>
<td>70-99</td>
<td>1419.72-2822.8</td>
</tr>
<tr>
<td></td>
<td>100-128</td>
<td>2,599.15-3,253.34</td>
</tr>
</tbody>
</table>

Fig. 7. Example of (2,4) sharing conditions to retrieve Lena image

3. Conclusion

In this paper a technique has been proposed to perform (k, n) secret image sharing based on wavelet transform in the case when the secret image is colored image. In the proposed system a compression of image is adapted to reduce the image size for efficient storage of images and transmission. A linear system has been constructed for share division. At the time of dividing an image into n number of shares a random number generator is introduced; a variable length key has been used to generate large prime numbers that was exploited to randomize the distribution of constructing values by the linear system into shares. A secret key is added to raise the robustness to the visual cryptography techniques, and the variable length of the key makes it more secure.

- Experimental results as shown in table 1 confirm that the proposed scheme gives smaller shadow size. Where at first the size of secret image is reduced approximately to 1/3 of its original size and second the
The size of each share is reduced to 1/q size of the reduced image and thus much storage space and transmission time is

- Table 2 shows the PSNR values of the reconstructed secret images range from 31.954 to 36.391 dB while the range of MSE is between 14.923 to 41.459.
- The security of the proposed system is guaranteed by several issues. The first one is each share depends on its own secret coefficients, a_i, which made the recovery secret image is too complicated for attacker. Second, the use of permutation key with time stamp for scrambling the secret image wavelet coefficient is too hard to be tracked. The third one is the use of diffusers to prune the existing bits significance in wavelet coefficients makes shares unbiased toward local significance; this will avoid the occurrence of localization problem.
- The security of the proposed system is guaranteed by several issues. The first one is each share depends on its own secret coefficients, a_i, which made the recovery secret image is too complicated for attacker. Second, the use of permutation key with time stamp for scrambling the secret image wavelet coefficient is too hard to be tracked. The third one is the use of diffusers to prune the existing bits significance in wavelet coefficients makes shares unbiased toward local significance; this will avoid the occurrence of localization problem.
- Experimental results as shown in Table 3 show that the time required to reconstruct secret image is increased with the increasing of coefficients values but this problem can be overcome by several key issues, one of these keys the coefficient can be chosen as small values, and the resulted shares can be encrypted individually using different secret keys to increase the security and make the brute force attack is more complicated.

References